

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Nguyễn Tuấn Anh

**PHÁT TRIỂN PHẦN MỀM NHẬN DẠNG HOA
TRÊN NỀN TẢNG THIẾT BỊ DI ĐỘNG**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ thông tin**

HÀ NỘI - 2019

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Nguyễn Tuấn Anh

**PHÁT TRIỂN PHẦN MỀM NHẬN DẠNG HOA
TRÊN NỀN TẢNG THIẾT BỊ DI ĐỘNG**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS. Nguyễn Thị Ngọc Diệp

Cán bộ đồng hướng dẫn: PGS.TS. Nguyễn Việt Hà

HÀ NỘI - 2019

**VIETNAM NATIONAL UNIVERSITY, HA NOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

Nguyen Tuan Anh

**DEVELOPING A FLOWER RECOGNITION
SOFTWARE ON MOBILE DEVICE PLATFORM**

**BACHELOR'S THESIS
Major: Information Technology**

Supervisor: Dr. Nguyen Thi Ngoc Diep

Co-Supervisor: Assoc. Prof. Nguyen Viet Ha

HANOI - 2019

LỜI CẢM ƠN

Lời đầu tiên cho tôi xin được gửi lời cảm ơn chân thành và sâu sắc nhất tới TS. Nguyễn Thị Ngọc Diệp, PGS. TS. Nguyễn Việt Hà, ThS. Nguyễn Ngọc Khuê những người đã hướng dẫn và chỉ bảo tận tình nhất cho tôi trong suốt quá trình hoàn thành khóa luận tốt nghiệp này.

Tôi xin được gửi lời cảm ơn tới toàn bộ các thầy giáo, cô giáo của trường Đại học Công Nghệ - Đại học Quốc Gia Hà Nội nhưng người đã tạo điều kiện tốt nhất để tôi có thể học tập, nghiên cứu và hơn cả là đã truyền thụ cho tôi những hành trang kiến thức đầy đủ nhất.

Tôi cũng xin gửi lời cảm ơn chân thành nhất tới các thành viên trong nhóm nghiên cứu SkyLab, tới những người bạn người anh, chị đã giúp đỡ tôi hoàn thiện cả về kiến thức chuyên môn và kỹ năng học tập nghiên cứu.

Cuối cùng và không thể thiếu đó là lời cảm ơn tới bố mẹ và chị tôi và đặc biệt là bạn Dung Phùng những người đã luôn bên cạnh tôi giúp đỡ và động viên cổ vũ tinh thần tôi trong những lúc khó khăn nhất.

Tôi xin chân thành cảm ơn!

Hà Nội, ngày 24 tháng 04 năm 2019
Sinh viên

Nguyễn Tuấn Anh

TÓM TẮT

Khóa luận này trình bày một ứng dụng nhận dạng tên loài hoa dựa vào ảnh đầu vào trên nền tảng di động. Ứng dụng này được xây dựng với ba mô hình học máy chính: (1) một mô hình phân loại nhị phân để phát hiện có đối tượng hoa trong ảnh hay không, (2) một mô hình nhận dạng tên của loài hoa trong ảnh, và (3) một thuật toán tìm kiếm ảnh tương tự khi hiển thị các ảnh kết quả.

Ứng dụng được phát triển sử dụng phương pháp sử dụng lại một mô hình học sâu đã được huấn luyện trên bộ dữ liệu lớn khác để trích xuất dữ liệu của ảnh. Thực nghiệm so sánh phương pháp tiếp cận này với các cách trích xuất thuộc tính truyền thống chỉ ra tính khả thi và tiết kiệm của việc sử dụng lại mô hình giữa các bài toán nghiên cứu trong lĩnh vực xử lý ảnh.

Những đóng góp chính của khóa luận là: (1) Việt hoá tên cũng như đặc điểm về sinh trưởng, cách trồng của 102 loài hoa trong bộ dữ liệu Oxford-102, (2) phát triển ứng dụng trên nền tảng di động, và (3) phát triển mô hình phân loại ảnh có hoa hay không với độ chính xác 98.6% và mô hình nhận dạng tên loài hoa với độ chính xác là 78.56%.

Từ khóa: nhận diện hoa, phát hiện hoa, ứng dụng di động

ABSTRACT

This thesis presents an recognition flowers' names application based on the input images on the mobile platform. This application is built with three main machine learning models: (1) a binary classification model to detect flower in an image, (2) a model that recognizes the name of the flower in an image, and (3) an algorithm that judges the similarity of visual content of an image in each class.

The application is developed using a method of reusing a deep learning model trained by another large dataset to extract features of an image. The comparison between this approach and the traditional feature extraction methods points to the feasibility and savings of reusing model of research problems in the field of image processing.

The main contributions of this thesis are: (1) Translating names and the growth characteristics of plants as well as methods of planting of 102 flower species in Oxford-102 dataset, (2) Developing an application on mobile platform, and (3) Developing a flowers detection model with 98.6% accuracy and an recognition model of flowers' names with 78.56% accuracy.

Keywords: *flower detection, flower recognition, mobile application*

LỜI CAM ĐOAN

Tôi xin cam đoan toàn bộ khóa luận về ứng dụng phát triển phần mềm nhận dạng hoa trên thiết bị di động bao gồm mô hình nhận diện, phần mềm di động và phần mềm hệ thống là do tôi thực hiện dưới sự hướng dẫn của TS. Nguyễn Thị Ngọc Diệp và PGS. TS. Nguyễn Việt Hà. Tất cả các công trình nghiên cứu, bài báo, khóa luận, tài liệu của các tác giả khác được tôi sử dụng trong khóa luận này đều được trích dẫn tường minh và đều có trong danh sách tài liệu tham khảo.

Hà Nội, ngày 24 tháng 04 năm 2019
Sinh viên

Nguyễn Tuấn Anh

Mục lục

Danh sách bảng	viii
Danh sách ký hiệu, chữ viết tắt	ix
Danh sách hình vẽ	x
Chương 1 Mở đầu	1
1.1 Bối cảnh nghiên cứu	1
1.2 Mô tả bài toán nhận dạng hoa	2
1.3 Mục tiêu và phương pháp giải quyết vấn đề	4
1.3.1 Mục tiêu của khóa luận	4
1.3.2 Phương pháp giải quyết bài toán nhận dạng hoa	4
1.3.3 Những điểm mạnh của phương pháp này	5
1.4 Bố cục các phần trong khóa luận	5
Chương 2 Nghiên cứu liên quan đến bài toán nhận dạng hoa	6
2.1 Các bộ dữ liệu hoa	6
2.2 Các nghiên cứu phân loại hoa sử dụng bộ hoa của Oxford-102	7
2.2.1 Các nghiên cứu về tiền xử lý ảnh trước khi trích xuất thông tin	7
2.2.2 Các nghiên cứu về nhận dạng dựa trên vector đặc trưng	9
2.2.3 Nghiên cứu về trích xuất thông tin từ ảnh gốc	11

2.2.4	Nghiên cứu về nhận dạng ảnh dựa trên tìm kiếm	13
2.3	Các ứng dụng di động cho nhận dạng hoa	15
Chương 3	Phát triển phần mềm nhận dạng hoa trên thiết bị di động	18
3.1	Mô tả bài toán	18
3.2	Mô tả mô hình hệ thống	20
3.2.1	Mô hình chung	20
3.2.2	Mô hình phát hiện hoa	24
3.2.3	Mô hình phân loại hoa	26
3.3	Xây dựng và cài đặt ứng dụng trên thiết bị di động	28
3.3.1	Mô tả sơ đồ các ca sử dụng chính của ứng dụng	28
3.3.2	Giao diện của ứng dụng trên điện thoại di động	29
Chương 4	Kết quả thực nghiệm	36
4.1	Xây dựng bộ dữ liệu	36
4.1.1	Bộ dữ liệu dùng để phát hiện đối tượng hoa trong ảnh.	36
4.1.2	Bộ dữ liệu dùng để phân loại tên loài hoa	39
4.1.3	Bộ dữ liệu ảnh chụp thực tế.	40
4.2	Môi trường thực nghiệm	41
4.3	Kết quả của mô hình phát hiện hoa.	42
4.4	Kết quả của mô hình phân loại hoa	44
4.4.1	Kết quả sử dụng bộ test Oxford-102	44
4.4.2	Giải thích các thí nghiệm	44
4.5	Kết quả sử dụng bộ test ảnh chụp thực tế	50
4.6	So sánh kết quả với các ứng dụng đã có sử dụng bộ ảnh chụp thực tế .	52
4.7	So sánh kết quả tìm kiếm dùng Google Image	54

4.8	Thảo luận về các kết quả	56
Chương 5	Kết luận	58

Danh sách bảng

2.1	Bảng kết quả của nghiên cứu [2]	9
2.2	Bảng kết quả của nghiên cứu [4]	12
4.1	Bảng danh sách các loài hoa phổ biến ở Việt Nam có trong bộ dữ liệu ảnh chụp thực tế	41
4.2	Bảng kết quả phân loại của mô hình phát hiện hoa	42
4.3	Bảng kết quả độ chính xác phân loại hoa	44
4.4	Bảng kết quả phân loại hoa sử dụng phương pháp KNN	45
4.5	Bảng kết quả phân loại hoa sử dụng hàm SVC	47
4.6	Bảng kết quả phân loại hoa sử dụng hàm LinearSVC	49
4.7	Bảng mô tả các mô hình phân loại bổ sung	50
4.8	Bảng so sánh kết quả phân loại hoa trên bộ ảnh thực tế	51
4.9	Bảng so sánh kết quả phân loại hoa của các ứng dụng sử dụng bộ ảnh chụp thực tế	53
4.10	Bảng so sánh kết quả tìm kiếm bằng Google Image	55

Danh sách ký hiệu, chữ viết tắt

CNN	Convolutional Neural Network
CNNaug	Convolutional Neural Network + image augmentation
HOG	Histogram of Gradients
HSV	Hue, saturation, value
KNN	K-Nearest Neighbour
LinearSVC	Linear Support Vector Classification
MRF	Markov Random Field
ONE	Online Nearest-neighbor Estimation
PCA	Principal Component Analysis
RGB	Red, green, blue
SIFT	scale-invariant feature transform
SVC	Support Vector Classification
SVM	Support-vector machine

Danh sách hình vẽ

1.1	Ví dụ về tìm kiếm hình ảnh của Google Images	2
1.2	Hình ảnh ví dụ về các loài hoa có cấu trúc giống nhau trong bộ dữ liệu Oxford-102	3
1.3	Hình ảnh ví dụ về các loài hoa thay đổi hình dạng theo thời gian trong bộ dữ liệu Oxford-102	3
2.1	Hình ảnh trong nghiên cứu [1] về phương pháp tách phông nền	8
2.2	Hình ảnh trong nghiên cứu [1] về các loài hoa có hình dáng tương tự nhau	10
2.3	Hình ảnh trong nghiên cứu [1] về các loài hoa đặc trưng về hoa văn . .	11
2.4	Hình ảnh mô tả kiến trúc của VGG19 [15] lấy từ trang researchgate.net	13
2.5	Hình ảnh tóm tắt về phương pháp ước lượng láng giềng gần nhất trong nghiên cứu [3] của tác giả	14
2.6	Hình ảnh đại diện của các ứng dụng	15
2.7	Hình ảnh kết quả toàn tiếng Anh khi tra cứu ảnh hoa hướng dương bằng các ứng dụng	16
2.8	Hình ảnh về ứng dụng PlantFinder cần phải khoanh vùng phần có hoa bằng tay	17
2.9	Hình ảnh về ứng dụng chưa có chức năng phát hiện hoa trong ảnh . .	17

3.1	Ví dụ về đầu vào và đầu ra của bài toán	19
3.2	Mô hình chung của hệ thống	20
3.3	Ví dụ về bước tách phông nền	21
3.4	Mô hình phát hiện hoa	24
3.5	Mô hình phân loại hoa	26
3.6	Hình ảnh về các ca sử dụng của ứng dụng	28
3.7	Hình ảnh giao diện tra cứu và kết quả tra cứu của ứng dụng.	30
3.8	Hình ảnh giao diện của ứng dụng khi không phát hiện hoa trong ảnh. .	31
3.9	Hình ảnh giao diện tìm kiếm bằng tên.	32
3.10	Hình ảnh giao diện chi tiết loài hoa.	33
3.11	Hình ảnh giao diện xem lại các ảnh của bản thân và mọi người.	34
3.12	Hình ảnh giao diện xem nhận xét của mọi người.	35
4.1	Hình ảnh lấy trong bộ ảnh không có hoa	37
4.2	Hình ảnh lấy trong bộ ảnh có hoa	38
4.3	Hình ảnh về trang web chứa tên loài hoa của bộ dữ liệu Oxford-102 .	40
4.4	Hình ảnh ví dụ trường hợp nhận dạng đúng	56
4.5	Hình ảnh ví dụ về trường hợp nhận dạng sai	57

Chương 1

Mở đầu

1.1 Bối cảnh nghiên cứu

Cùng với sự phát triển của công nghệ nói chung cũng như sự phát triển của các thiết bị di động nói riêng ta có thể nhận thấy một điều rằng ngày nay điện thoại di động là một dụng cụ luôn đi kèm mỗi người mỗi khi đi ra khỏi nhà, đi làm hay đi chơi.

Khi muốn tìm kiếm tên của một bông hoa bạn có thể sử dụng thông qua công cụ tìm kiếm Google Images nhưng không thật may mắn các kết quả tìm kiếm chỉ là các hình ảnh tương tự với ảnh mà bạn chụp và tên các loài hoa thì toàn là tiếng Anh và tiếng Latin, không những thế đôi khi các kết quả trả về từ Google lại không hề liên quan tới tên loài hoa đó. Ngay cả các ứng dụng nhận dạng hoa đang có sẵn hiện tại cũng không có một ứng nào có hiển thị tên các loài hoa bằng tiếng Việt. Điều này sẽ là một bất lợi khá lớn đối với một số người và nhất là với học sinh khi muốn tìm hiểu về các loài hoa. Do vậy cần một ứng dụng có chức năng nhận dạng tên loài hoa từ các ảnh chụp trên thiết bị di động được xây dựng với dữ liệu tên các loài hoa bằng tiếng Việt với mục đích chính là đáp ứng được nhu cầu tìm kiếm về các loài hoa dành cho người Việt.

Floribunda (rose) - Wikipedia

[https://en.wikipedia.org/wiki/Floribunda_\(rose\)](https://en.wikipedia.org/wiki/Floribunda_(rose)) ▾

Floribunda (Latin for "many-flowering") is a modern group of garden roses that were developed by crossing hybrid teas with polyantha roses, the latter being ...

About 25,270,000,000 results (1.98 seconds)

15 Floribunda Roses for the Flower Garden -

<https://www.thespruce.com/outdoors-gardening/garden-roses-136070>

Feb 28, 2019 - Gene Boerner and Betty Prior Floribunda Rose palette of the hybrid tea rose, and you get floribunda roses.



Image size:
225 × 225

Find other sizes of this image:
[All sizes](#) - [Medium](#)

Possible related search: [nepal republic day 2017](#)

Visually similar images



Nepal Republic Day 2017 - Google

www.google.com/doodles/nepal-republic-day-2017 ▾

May 29, 2017 - The **Republic** of Nepal is a nation of immense diversity and is the only country in the world whose flag is not rectangular!

Nepal Republic Day 2017 - YouTube

<https://www.youtube.com/watch?v=i9eyPcvvN1k>

May 28, 2017 - Nepal Republic Day 2017. Doodle Videos. Load now. Please try again later. Published on May 28, 2017 ...

Hình 1.1: Ví dụ về tìm kiếm hình ảnh của Google Images

1.2 Mô tả bài toán nhận dạng hoa

Mô tả bài toán: bài toán nhận dạng hoa thông qua ảnh chụp sẽ có đầu vào là một hình ảnh được ghi lại từ các thiết bị ghi hình và đầu ra có thể có hai trường hợp

- Trường hợp 1: Trong ảnh không có hoa thì kết quả trả về là không tồn tại hoa.
- Trường hợp 2: Trong ảnh có hoa thì kết quả trả về là một danh sách 5 loài hoa có kết quả nhận dạng tốt nhất.

Những khó khăn khi nhận diện hoa

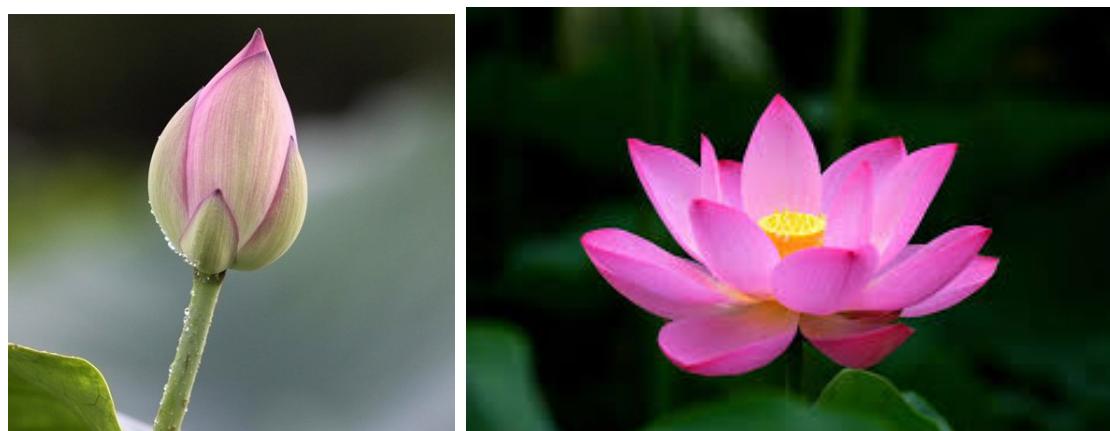
- Các loài mọc trong tự nhiên có số lượng loài rất lớn với vô số màu sắc, kiểu dáng hình thù khác nhau gây khó khăn khi phải phân loại với một số lượng loài lớn.
- Các loài hoa đôi khi có kiểu dáng tương đối giống nhau.

- Các loài hoa trong thời gian sinh trưởng có rất nhiều hình thái khác nhau gây ra nhầm lẫn một ví dụ điển hình có thể thấy đó là hoa sen khi chưa nở và lúc đã nở có hình dạng rất khác nhau.



Hình 1.2: Hình ảnh ví dụ về các loài hoa có cấu trúc giống nhau trong bộ dữ liệu Oxford-102

(Ở bên trái là ảnh hoa hướng dương còn ở bên phải là ảnh hoa cúc)



Hình 1.3: Hình ảnh ví dụ về các loài hoa thay đổi hình dạng theo thời gian trong bộ dữ liệu Oxford-102

(Cả hai bức ảnh đều là hoa sen trong đó bên trái là ảnh khi chưa nở còn ở bên phải là khi đã nở)

1.3 Mục tiêu và phương pháp giải quyết vấn đề

1.3.1 Mục tiêu của khóa luận

Mục tiêu chính của khóa luận này muốn nhắm tới đó là phát triển một ứng dụng nhận dạng tên các loài hoa dành cho người Việt trên nền tảng thiết bị di động. Ứng dụng sẽ không đơn thuần chỉ là một công cụ tra cứu mà còn có ý nghĩa trở thành một công cụ học tập, nghiên cứu về thiên nhiên và đặc biệt là các loài hoa giúp các em học sinh, mọi người ở mọi lứa tuổi có thêm một công cụ hữu ích trong việc học tập và nghiên cứu về các loài hoa. Hơn nữa là qua ứng dụng này để xuất phương pháp sử dụng mạng học sâu đã được huấn luyện VGG19 [15] trong các mô hình trích xuất thuộc tính từ ảnh, đồng thời tạo tiền đề phát triển cho những nghiên cứu sau này về đề tài nhận diện ảnh.

1.3.2 Phương pháp giải quyết bài toán nhận dạng hoa

Để giải quyết được bài toán nhận dạng tên loài hoa sử dụng ảnh ta cần phải qua năm bước chính đó là:

- **Bước 1:** Tiền xử lý dữ liệu hình ảnh.
- **Bước 2:** Trích xuất các thuộc tính từ ảnh và chuyển thành các vector thuộc tính.
- **Bước 3:** Sử dụng các thuật toán phân loại để phát hiện có hoa trong ảnh hay không.
- **Bước 4:** Nếu phát hiện thấy hoa trong ảnh thì sử dụng vector thuộc tính đó để nhận dạng tên loài hoa.
- **Bước 5:** Tính toán độ tương tự của các ảnh trong từng loài hoa để đưa ra các ảnh phù hợp nhất cho kết quả.

1.3.3 Những điểm mạnh của phương pháp này

Qua thực nghiệm và so sánh phương pháp trích xuất thuộc tính sử dụng mô hình học sâu VGG19 [15] với các cách trích xuất thuộc tính truyền thống có kết quả khá khả quan và có tính khả thi cao.

Hơn nữa việc sử dụng lại mô hình trích xuất thuộc tính giữa các bài toán nhận dạng sẽ tiết kiệm được đáng kể công sức cũng như cải thiện được độ chính xác phân loại.

1.4 Bố cục các phần trong khóa luận

Các phần trong khóa luận được cấu trúc như sau.

- **Chương 1:** Giới thiệu về bài toán nhận dạng hoa, mục đích và các phương pháp sử dụng để giải quyết bài toán phát triển ứng dụng nhận dạng hoa trên thiết bị di động.
- **Chương 2:** Trình bày các nghiên cứu liên quan đến bài toán nhận dạng hoa. Các phương pháp trích xuất thuộc tính truyền thống và các phương pháp mới.
- **Chương 3:** Trình bày và giải thích về mô hình tổng quát của ứng dụng cũng như cách huấn luyện các mô hình phân loại được sử dụng trong hệ thống.
- **Chương 4:** Mô tả về cách thu thập dữ liệu, cách cài đặt và sử dụng các thuật toán để giải quyết vấn đề đồng thời giải thích về mục đích cũng như đưa ra các số liệu thống kê về các kết quả đã đạt được trong từng thí nghiệm.
- **Chương 5:** Dưa ra các kết luận về những vấn đề đã giải quyết và những vấn đề chưa giải quyết được đồng thời giới thiệu hướng nghiên cứu để tiếp tục phát triển trong tương lai.

Chương 2

Nghiên cứu liên quan đến bài toán nhận dạng hoa

2.1 Các bộ dữ liệu hoa

Có khá nhiều các bộ dữ liệu về hoa nổi tiếng và đa dạng ta có thể kể đến như

- Bộ dữ liệu về hoa của ImageNet [17].
- Oxford-17 [1] bao gồm 1360 ảnh thuộc 17 loài hoa.
- Oxford-102 [1] bao gồm 8189 ảnh thuộc 102 loài hoa.
- Kaggle flower dataset [18] bao gồm 4242 ảnh của 5 loài hoa.
- Bộ dữ liệu về thực vật trên trang plant database garden.org.

Trong khóa luận này sử dụng bộ dữ liệu hoa Oxford-102 [1], vì các lý do sau.

- Số lượng loài nhiều, số lượng ảnh mỗi loài cũng nhiều.
- Các ảnh chụp đều là ảnh chụp cận cảnh bông hoa và đa dạng về góc chụp, ánh sáng.

- Các ảnh mỗi loài đều có số lượng loại phong phú, ví dụ ở đây hoa hồng có cả đỏ, hồng, trắng.. thay vì chỉ một loại hoa hồng đỏ.
- Có nhiều nghiên cứu sử dụng bộ, cụ thể trong khóa luận có đề cập tới 4 nghiên cứu [1] [2] [3] [4].

2.2 Các nghiên cứu phân loại hoa sử dụng bộ hoa của Oxford-102

Các nghiên cứu về phân loại hoa [1] [2] [3] [4] có sử dụng bộ dữ liệu hoa Oxford-102 [1] trong đó có thể nhóm lại thành các nhóm theo các phương pháp chính mà các nghiên cứu có sử dụng.

- Tiền xử lý ảnh trước khi trích xuất thông tin.
- Nhận dạng dựa trên vector đặc trưng.
- Trích xuất thông tin từ ảnh gốc.
- Nhận dạng dựa trên tìm kiếm (retrieval-based).

2.2.1 Các nghiên cứu về tiền xử lý ảnh trước khi trích xuất thông tin

Dầu vào của hệ thống nhận dạng hoa là một bức ảnh được thu lại từ một máy ảnh hoặc các thiết bị ghi hình khác. Các ảnh này đôi khi có chất lượng không như mong muốn, ví dụ như quá tối, chủ thể quá nhỏ, quá to. Nguyên nhân chủ yếu có thể do thiết bị ghi hình không tốt hoặc điều kiện ghi hình không tốt ví dụ trời quá tối, chụp ảnh trong khi đang di chuyển, người chụp ảnh không chú ý. Do vậy để bước phân loại có được kết quả tốt nhất ta cần phải có một bước tiền xử lý ảnh.

Trong nghiên cứu [1] [2] các tác giả sử dụng phương pháp tách phông nền MRF [8]. Theo như hình 2.1 được lấy trong nghiên cứu [1] các ảnh bên trên là ảnh gốc và

các ảnh phía dưới đã được tách phông nền. Ta có thể thấy ở bức ảnh thứ 2 bông hoa sau khi tách phông nền đã bị mất đi 1 phần, tỷ lệ mất mát đối tượng sau tách phông nền theo như báo cáo là 6%.



Hình 2.1: Hình ảnh trong nghiên cứu [1] về phương pháp tách phông nền

Bên cạnh đó còn một kỹ thuật nữa giúp cải thiện tập dữ liệu là tạo ra nhiều dữ liệu hơn từ dữ liệu ban đầu. Đối với xử lý ảnh, kỹ thuật này có tên là tăng cường ảnh (Image Augmentation) [10] [16]. Có thể giải thích như sau, chúng ta có một bông hoa hồng thì khi soi vào gương, dịch sang trái, phải, trên, dưới, đứng trong tối, ngoài sáng, phóng to, thu nhỏ, xoay 1 góc bất kỳ thì cũng vẫn là một bông hồng đó. Vậy thì từ một ảnh huấn luyện, ta có thể tạo ra vô vàn những ảnh tương tự và vẫn có thể được nhận ra là bông hồng bằng các phép biến đổi ảnh như: xoay ảnh, lật ảnh, dịch ảnh,... Và như vậy, chúng ta có một tập dữ liệu lớn hơn cho bước huấn luyện. Nghiên cứu [4] có sử dụng phương pháp này và đã tăng được độ chính xác từ 74,4% lên thành 86.8%.

2.2.2 Các nghiên cứu về nhận dạng dựa trên vector đặc trưng

Nghiên cứu [1] sử dụng phương pháp trích xuất các đặc trưng phổ biến của một bức ảnh bao gồm màu sắc, khối hình và hoa văn từ đó sử dụng phương pháp lảng giềng gần nhất để phân loại ra tên của loài hoa đó. Kết quả phân loại tốt nhất đạt được khi thử nghiệm trên bộ kiểm tra Oxford-17 là bằng cách kết hợp hình dạng và màu sắc đạt kết quả 81.3%.

Bảng 2.1 cho thấy các kết quả của nghiên cứu [2], tại đó các tác giả đã tính toán bốn đặc điểm khác nhau cho các bông hoa, cụ thể là hình dạng kết cấu cục bộ, hình dạng của ranh giới, phân bố không gian tổng thể của cánh hoa và màu sắc của các loài hoa. Sau đó kết hợp các tính năng bằng cách sử dụng thuật toán phân loại máy vector hỗ trợ SVM. Kết quả cho thấy độ chính xác phân loại cho các bộ thuộc tính khác nhau là khác nhau. Có thể thấy rằng việc kết hợp tất cả các thuộc tính mang lại độ chính xác tốt hơn nhiều so với sử dụng các thuộc tính đơn lẻ.

Bảng 2.1: Bảng kết quả của nghiên cứu [2]

Phương pháp trích xuất	Độ chính xác phân loại (%)
HSV	43.0
SIFT internal	55.1
SIFT boundary	32.0
HOG	49.6
HSV + SIFT internal	66.4
HSV + SIFT boundary	57.0
HSV + HOG	62.1
SIFT internal + HOG	66.4
SIFT boundary + HOG	55.3
HSV + SIFT internal + HOG	71.8
HSV + SIFT internal + SIFT boundary + HOG	72.8

Vector đặc trưng về khối hình

Ở hình 2.2 theo thứ tự từ trái qua phải: hoa gió, hoa buttercup và hoa thủy tiên.

Nhìn qua thì các loài hoa này có cấu trúc tương tự nhau nhưng lại không hoàn toàn giống nhau. Sử dụng các mô tả SIFT [6] các tác giả sẽ thu được một vector đặc trưng về khối hình, hình dáng của bông hoa có trong ảnh.



Hình 2.2: Hình ảnh trong nghiên cứu [1] về các loài hoa có hình dáng tương tự nhau

Vector đặc trưng về kết cấu trên vùng tiền cảnh sử dụng SIFT [6]

Các mô tả SIFT [6] được tính toán trên toàn bộ vùng hoa còn lại sau khi tách tiền cảnh. Kết quả thu được là một vector thuộc tính về kết cấu tiền cảnh gồm 128 chiều.

Vector đặc trưng kết cấu trên vùng ranh giới sử dụng SIFT [6]

Ranh giới của ảnh sau bước tiền xử lý tách phông nền cho ta ranh giới của hoa. Sau đó lấy mẫu các mô tả SIFT [6] trên đường viền của bông hoa, chúng ta có thể nhấn mạnh hơn đến hình dạng cục bộ của đường biên. Kết quả thu được cũng thu được một vector thuộc tính kết cấu nhưng trên vùng ranh giới bao gồm 128 chiều.

Vector đặc trưng về màu sắc

Một số hoa có màu sắc rất đặc trưng và có thể giúp thu hẹp đáng kể các loài để phân loại, nhưng nó sẽ không cho phép chúng ta xác định chính xác các loài hoa. Sử dụng mô tả màu bằng không gian màu HSV [1] để giảm hiệu ứng của ánh sáng môi trường tới ảnh.

Vector đặc trưng về hoa văn

Một số hoa có hoa văn đặc trưng trên cánh hoa của chúng. Những họa tiết này có thể đặc biệt hơn, chẳng hạn các sọc của hoa păng xê, các ô vuông ở hoa đầu rắn,

hoặc các chấm nhỏ ở hoa ly hổ các tác giả đã sử dụng bộ lọc MR8 [11] để trích xuất thuộc tính về hoa văn của bông hoa trong ảnh.



Hình 2.3: Hình ảnh trong nghiên cứu [1] về các loài hoa đặc trưng về hoa văn

Vector đặc trưng về phân bố tổng thể

Sử dụng phương pháp HOG [7], tương tự như các tính năng SIFT [6]. Tuy nhiên, thay vì được áp dụng cho các vùng cục bộ (bán kính R trong trường hợp SIFT[6]), HOG được áp dụng ở đây trên toàn bộ vùng hoa. Theo cách này, sẽ mô tả được sự phân bố không gian tổng thể của bông hoa trong bức ảnh.

2.2.3 Nghiên cứu về trích xuất thông tin từ ảnh gốc

Mạng tích chập OverFeat [4]

Trong nghiên cứu này [4] các tác giả sử dụng mạng tích chập có tên là OverFeat. Mô hình nhận đầu vào là hình ảnh màu có kích thước 221 x 221. OverFeat đã được đào tạo cho nhiệm vụ phân loại hình ảnh của ImageNet ILSVRC 2013 và thu được kết quả khá cao.

Các tác giả sử dụng bộ dữ liệu ImageNet ILSVRC13 [19] chứa 1,2 triệu hình ảnh được dán nhãn bằng tay của 1000 loại nhãn để huấn luyện cho mô hình. Sau đó có thử nghiệm trên bộ dữ liệu Oxford-102 [1] và thu được kết quả như bảng 2.2.

Đối với tất cả các thí nghiệm đều sử dụng đến lớp kết nối đầy đủ đầu tiên (lớp 22) của mạng để cho ra vector thuộc tính. Đối với tất cả các thử nghiệm, thay đổi kích thước toàn bộ hình ảnh thành 221 x 221. Đầu ra sẽ cho một vector thuộc tính

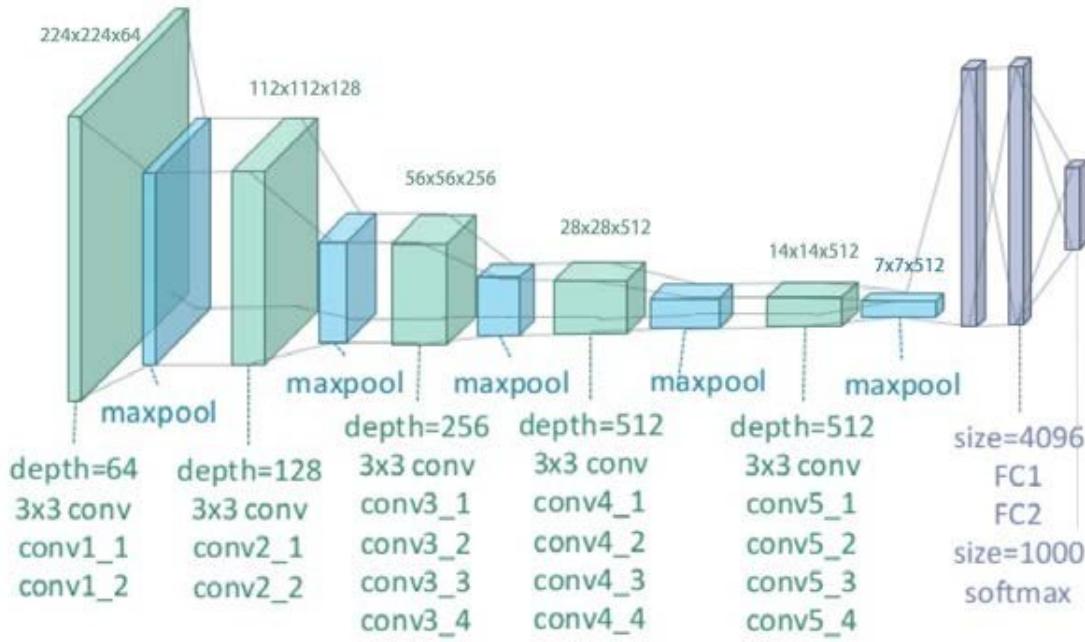
Bảng 2.2: Bảng kết quả của nghiên cứu [4]

Phương pháp	Độ chính xác phân loại (%)
CNN-SVM w/o seg	74.7
CNNaug-SVM w/o seg	86.8

có 4096 chiều. Trong nghiên cứu có đề cập tới hai loại cài đặt là có sử dụng phương pháp tăng cường ảnh CNNaug và không sử dụng phương pháp tăng cường ảnh. Đối với các nhu cầu phân loại trong đó các nhãn không loại trừ lẫn nhau các tác giả sử dụng chiến lược một với tất cả (one-against-all), và trong phần còn lại của các thí nghiệm sử dụng các SVM tuyến tính một đối một (one-against-one).

Mạng tích chập VGG19 [15]

Dựa trên các công trình nghiên cứu trước đó thì vào năm 2015 có một nghiên cứu [15] đề xuất sử dụng mạng tích chập ConvNet [15] gồm rất nhiều lớp để có thể đáp ứng được khả năng phân loại trên diện rộng. Trong nghiên cứu này, các tác giả đã điều tra ảnh hưởng của độ sâu mạng tích chập đến độ chính xác trong việc nhận dạng hình ảnh với quy mô lớn. Qua đánh giá kỹ lưỡng các mạng đã được tăng độ sâu các tác giả đã đạt được sự cải thiện đáng kể khi nâng độ sâu lên 16 đến 19 lớp. Trong hình 2.4 là mô tả cách cài đặt cho mạng ConvNet. Kết quả thu được là một vector thuộc tính 4096 chiều.



Hình 2.4: Hình ảnh mô tả kiến trúc của VGG19 [15] lấy từ trang researchgate.net

2.2.4 Nghiên cứu về nhận dạng ảnh dựa trên tìm kiếm

Phương pháp nhận dạng ảnh dựa trên tìm kiếm [20] có chút khác biệt so với bài toán phân loại hình ảnh, đầu vào của bài toán vẫn là một hình ảnh và nhiệm vụ đặt ra ở đây là phải sắp xếp, hoặc tìm kiếm, những hình ảnh có liên quan trong cơ sở dữ liệu. Có hai hướng tiếp cận trong bài toán truy hồi là theo hướng khái niệm và hướng nội dung.

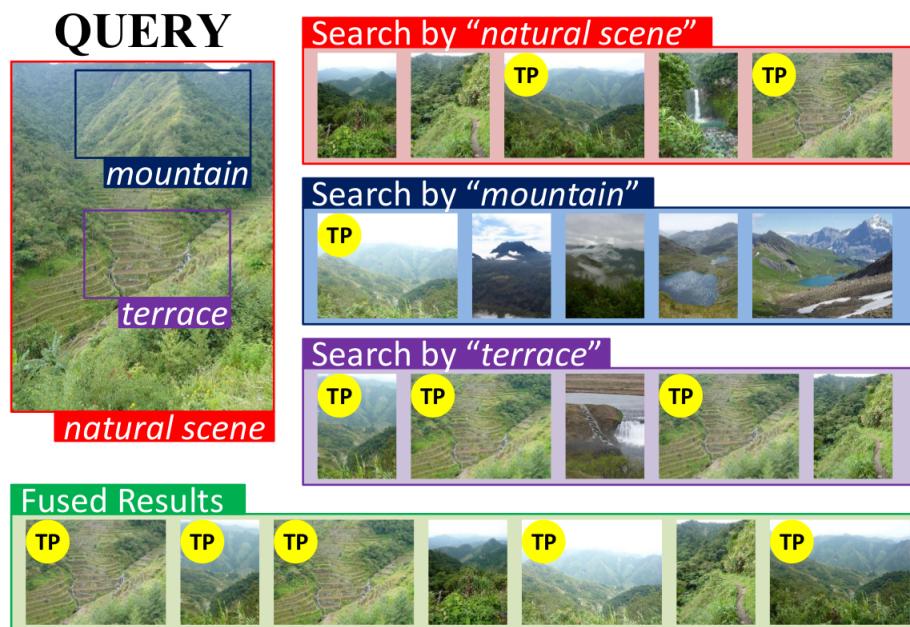
- Tiếp cận theo hướng khái niệm: là việc tìm kiếm dựa trên các thông tin liên quan đến một bức ảnh như tên bức ảnh, tiêu đề, chủ đề hoặc phần chữ miêu tả về bức ảnh đó.
- Tiếp cận theo hướng nội dung: là việc tìm kiếm dựa trên nội dung của ảnh.

Nghiên cứu về nhận dạng hoa có sử dụng phương pháp tìm kiếm hình ảnh [3] tại đây các tác giả sử dụng thuật toán ước lượng láng giềng gần nhất trực tiếp (Online Nearest neighbor Estimation - ONE) cho cả phân loại và truy hồi hình ảnh, bằng cách tính độ tương tự giữa các đối tượng trong ảnh dùng để tìm kiếm với từng lớp hoặc

các hình ảnh của lớp đó. Kết quả phân loại tốt nhất mà các tác giả đạt được trên bộ dữ liệu Oxford-102 [1] là 86.24%

Các bước thực hiện của phương pháp

Đầu tiên sử dụng phương pháp phát hiện đa vật thể dựa theo các nghiên cứu [12] [13] [14] trên hình ảnh truy vấn và từng hình ảnh của các lớp. Ở đây hình ảnh bên trái ở trên là hình ảnh cần tìm kiếm ta sẽ sử dụng phương pháp phát hiện đa vật thể ra được các vật thể có trong bức ảnh là "mountain", "terrace" và "natural scene". Trong bộ dữ liệu của ta, ta sẽ lấy ra các lớp về "mountain", "terrace" và "natural scene". Rồi sau đó trích xuất các thuộc tính trên mỗi đối tượng đó để cung cấp một mô tả sát nhất cho hình ảnh tìm kiếm. So sánh mức độ liên quan giữa các đối tượng đã được phát hiện trong ảnh truy vấn với các đối tượng trong các ảnh thành viên trong lớp đó. Các ảnh nào có độ tương tự gần nhất với các đối tượng có mặt trong hình ảnh cần tìm kiếm sẽ có chữ "TP-true positive" màu vàng. Trong bước tìm kiếm, mức độ liên quan của hình ảnh cần tìm kiếm đối với một danh mục hoặc hình ảnh ứng cử viên được tính bằng mức trung bình gần nhất khoảng cách từ hình ảnh cần tìm kiếm đối tượng đến các đối tượng trong lớp đó.



Hình 2.5: Hình ảnh tóm tắt về phương pháp ước lượng láng giềng gần nhất trong nghiên cứu [3] của tác giả

2.3 Các ứng dụng di động cho nhận dạng hoa

Các ứng dụng nhận dạng hoa và thực vật có xuất hiện khá nhiều nhưng lại có sự khác biệt khá lớn về độ chính xác và tốc độ nhận dạng. Trong khóa luận này lấy ra 3 ví dụ về ứng dụng nhận dạng hoa có độ chính xác và tốc độ nhận dạng khá tốt.

- PictureThis - Flower & Plant Identification: do công ty Glority Software Limited.
- PlantNet Plant Identification: do công ty plantnet-project.org.
- PlantFinder - Flower & Plant Identification: do công ty HD Wallpapers Production.



Hình 2.6: Hình ảnh đại diện của các ứng dụng

(Lần lượt từ trái qua phải ta có các ứng dụng là PictureThis, PlantNet, PlantFinder)

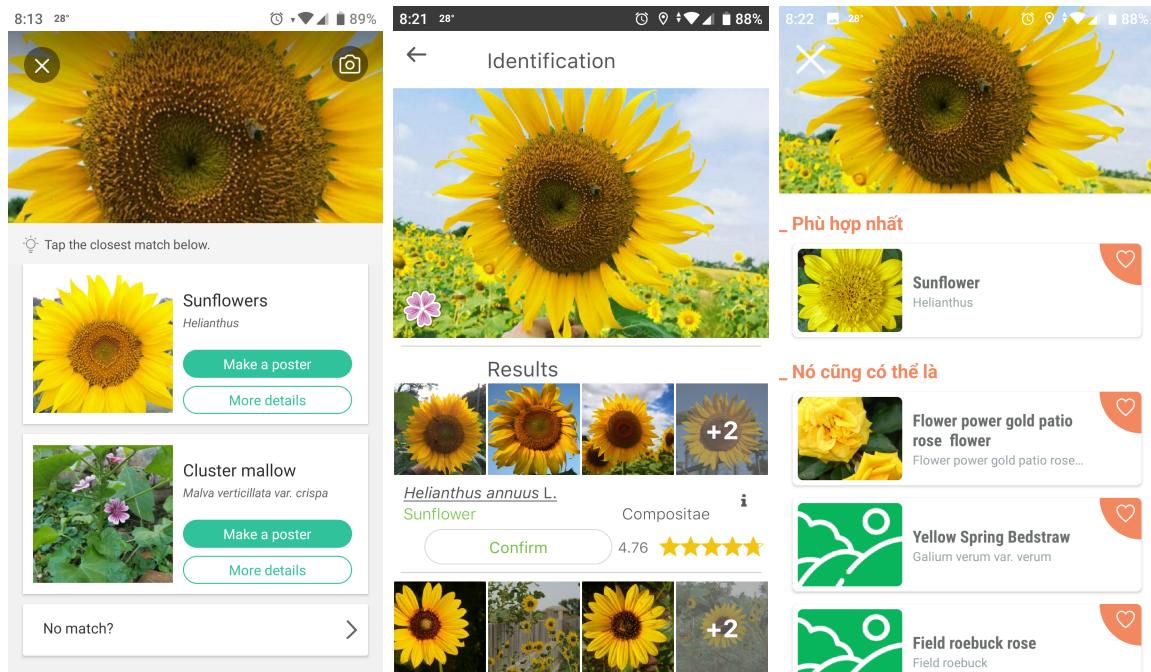
Ưu điểm chung của các ứng dụng:

- Các ứng dụng có độ nhận dạng chính xác khá tốt.
- Thời gian phản hồi ở mức khá.
- Có nêu ra đặc điểm chi tiết của loài hoa.

Một số nhược điểm còn tồn tại:

- Toàn bộ tên loài hoa và chi tiết về đặc điểm của loài hoa đang ở tiếng anh và tiếng latin, gây ra đôi chút khó khăn cho người Việt khi cần tra cứu.

- Một số ứng dụng sau khi chụp xong cần phải khoanh vùng hoa cần tìm bằng tay, chưa tự động phát hiện.
- Một số ứng dụng chưa có bước phát hiện xem ảnh có hoa hay không.

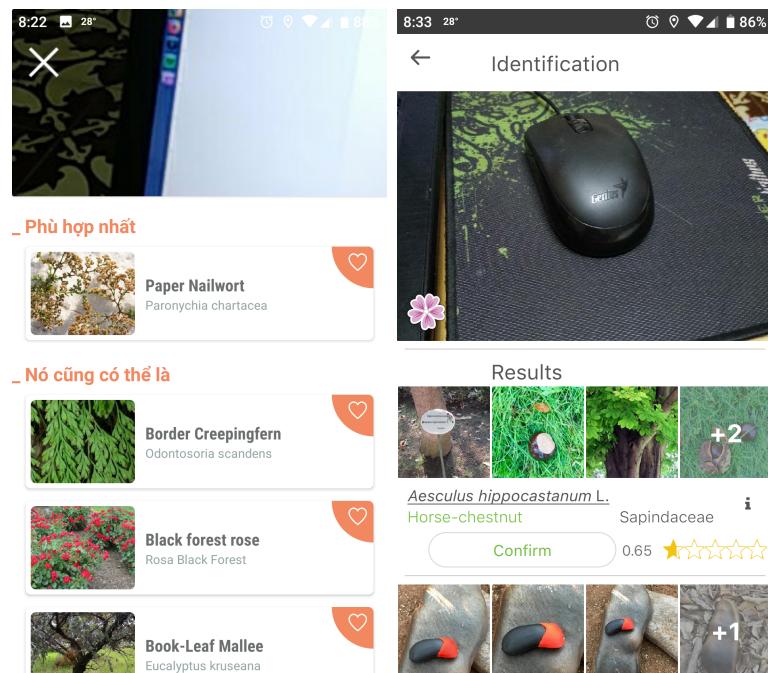


Hình 2.7: Hình ảnh kết quả toàn tiếng Anh khi tra cứu ảnh hoa hướng dương bằng các ứng dụng

(Lần lượt từ trái qua phải là ảnh chụp màn hình kết quả của các ứng dụng PlantFinder, PlantNet và cuối cùng là Picture this.)



Hình 2.8: Hình ảnh về ứng dụng PlantFinder cần phải khoanh vùng phần có hoa bằng tay



Hình 2.9: Hình ảnh về ứng dụng chưa có chức năng phát hiện hoa trong ảnh

Chương 3

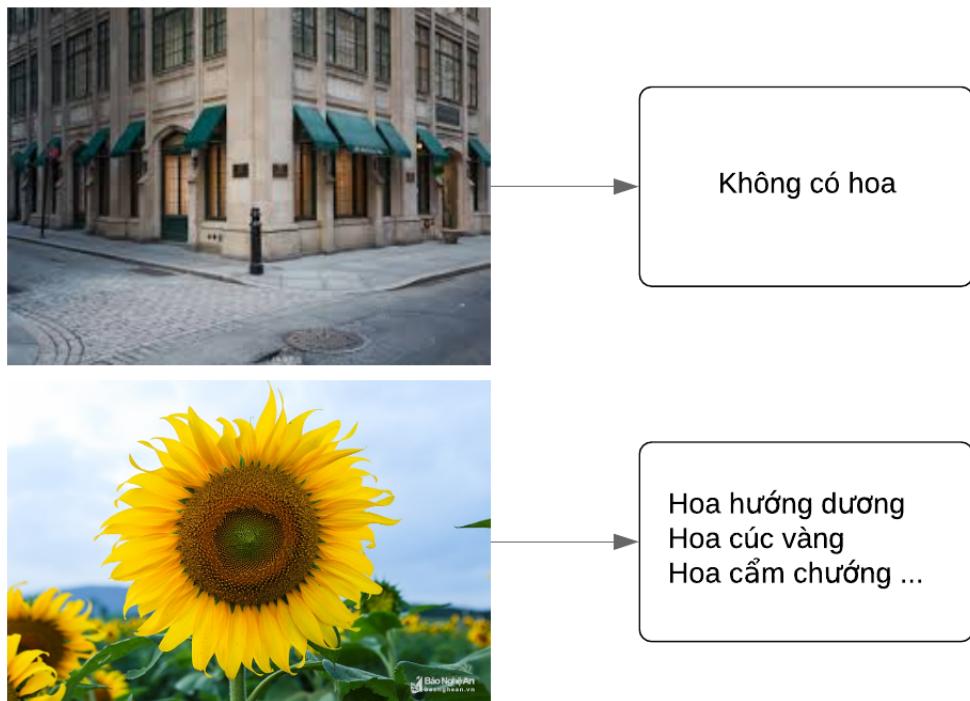
Phát triển phần mềm nhận dạng hoa trên thiết bị di động

Như đã trình bày tại **chương 2**, bước trích xuất thông tin từ ảnh là bước rất quan trọng trong mô hình nhận diện ảnh làm tốt bước này ta sẽ cải thiện được đáng kể độ chính xác và hiện nay có rất nhiều phương pháp để trích xuất thông tin từ ảnh. Để tăng cường độ chính xác nhận diện ảnh cho ứng dụng tôi có thêm một bước vào trước bước phân loại hoa đó là phát hiện ảnh có hoa hay không. Qua sự tham khảo và học hỏi của các nghiên cứu đã được đề cập trong chương 2, ngoài ra tôi còn học hỏi những điểm mạnh và tìm cách khắc phục những điểm yếu còn tồn tại của các ứng dụng đã có để đề xuất phương pháp giải quyết vấn đề cho ứng dụng của tôi.

3.1 Mô tả bài toán

Ứng dụng nhận dạng hoa trong khóa luận này sẽ có chức năng chính đó là nhận dạng hoa thông qua ảnh chụp. Như vậy đầu vào và đầu ra của bài toán sẽ là một hình ảnh được ghi lại từ các thiết bị ghi hình và đầu ra có thể có hai trường hợp

- Trường hợp 1: Trong ảnh không có hoa thì kết quả trả về là không tồn tại hoa.
- Trường hợp 2: Trong ảnh có hoa thì kết quả trả về là một danh sách 5 loài hoa có kết quả nhận dạng tốt nhất.

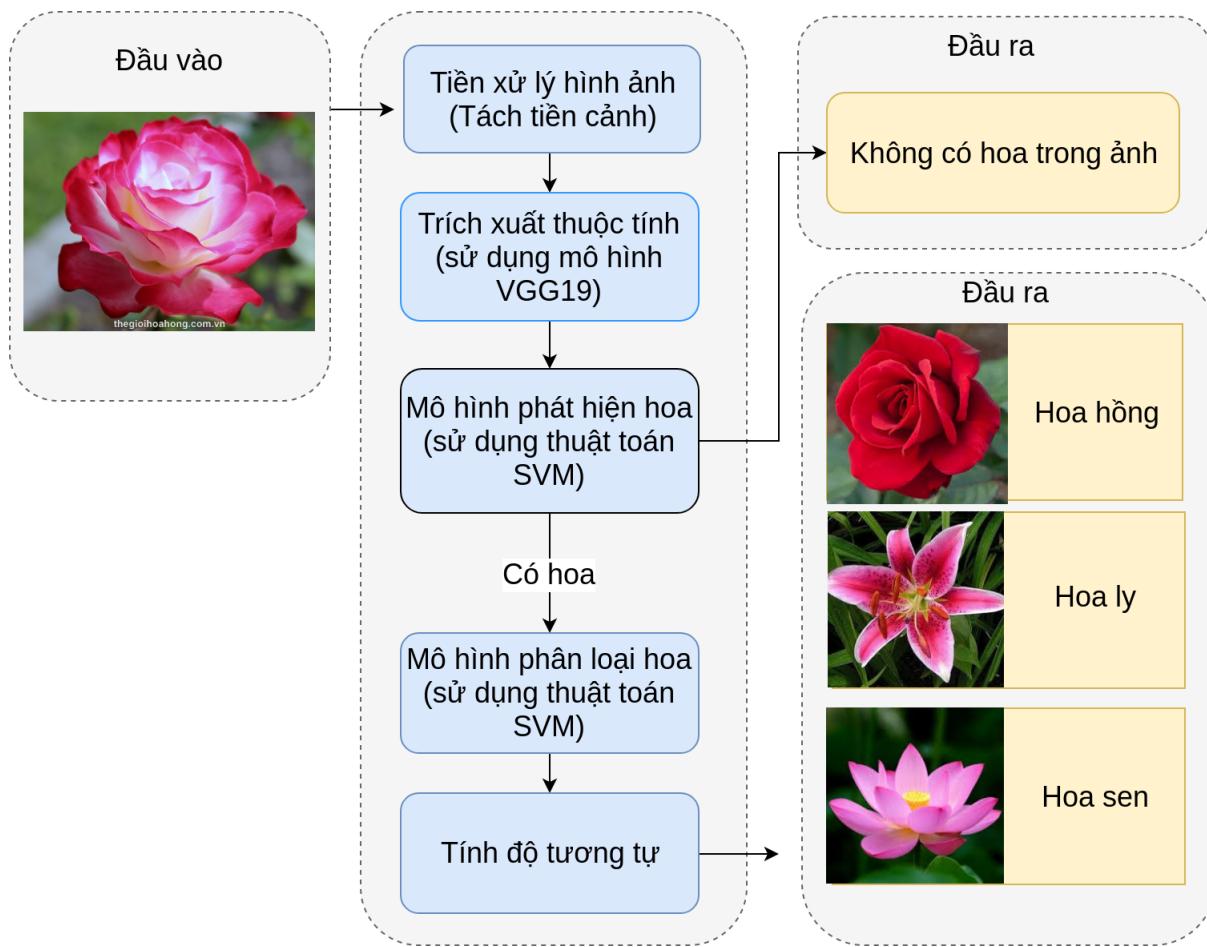


Hình 3.1: Ví dụ về đầu vào và đầu ra của bài toán

3.2 Mô tả mô hình hệ thống

Dưới đây là mô tả và giải thích chi tiết về các bước trong mô hình hệ thống qua sự tham khảo các nghiên cứu [1] [2] [3] [4] [15] và học hỏi về điểm mạnh và điểm yếu của các ứng dụng đã có về nhận dạng hoa trên thiết bị di động.

3.2.1 Mô hình chung



Hình 3.2: Mô hình chung của hệ thống

Bước 1: Tiền xử lý dữ liệu.

- Do các ảnh của người dùng tải lên hệ thống có thể gấp phải một số trường hợp

đặc biệt ví dụ như ảnh chụp bông hoa quá nhỏ, ảnh của hoa không nằm trong chính giữa bức ảnh ... Vì vậy, ảnh sau khi được chụp và gửi lên hệ thống sẽ được tiền xử lý bằng phương pháp tách phông nền để loại bỏ các phần không quan trọng.

- **Đầu vào:** Ảnh do người dùng chụp gửi lên hệ thống
- **Đầu ra:** Ảnh đã được tách phông nền.

```
1 import cv2
2
3 img = cv2.imread(img_path)
4 mask = np.zeros(img.shape[:2],np.uint8)
5 bgdModel = np.zeros((1,65),np.float64)
6 fgdModel = np.zeros((1,65),np.float64)
7 rect = (start_width,start_height,rect_width,rect_height)
8 cv2.grabCut(img,mask,rect,bgdModel,fgdModel,5,cv2.GC_INIT_WITH_RECT)
9 mask2 = np.where((mask==2)|(mask==0),0,1).astype('uint8')
```



Hình 3.3: Ví dụ về bước tách phông nền

Bước 2: Trích xuất thuộc tính.

- Đầu vào của bước này đó là bức ảnh mới có trọng tâm là chủ thể cần nhận dạng.
- Ảnh sẽ được thay đổi kích cỡ về dạng 224 x 224.
- Sử dụng mô hình VGG19 [15] để làm bước trích xuất thuộc tính. Chỉ sử dụng đến tầng thứ 19 (FC-4096) của mô hình và kết quả thu được ở đây là một vector thuộc tính có 4096 chiều.
- Vector thuộc tính sẽ được chuẩn hóa sẽ được chuẩn hóa L2 (L2 normalization).

```

1   from keras.applications.vgg19 import VGG19
2   from keras.preprocessing import image
3   from keras.applications.vgg19 import preprocess_input
4
5   base_model = VGG19(weights='imagenet')
6   model = Model(inputs=base_model.input,
7                 outputs=base_model.get_layer('fc2').output)
8   def get_feature_1_image(image_name):
9       img_path = './' + image_name + '.jpg'
10      img = image.load_img(img_path, target_size=(224, 224))
11      x = image.img_to_array(img)
12      x = np.expand_dims(x, axis=0)
13      x = preprocess_input(x)
14      features = model.predict(x)

```

Bước 3: Phát hiện hoa trong ảnh.

- Sử dụng phương pháp máy vector hỗ trợ để xây dựng mô hình phân loại nhị phân (binary classification).
- Đầu vào của bước này là vector thuộc tính đã được sinh ra ở bước 2.
- Nếu kết quả của mô hình phát hiện ảnh là không có hoa thì lập tức trả lại kết quả cho người dùng là không tìm thấy hoa.
- Nếu kết quả là có hoa thì chuyển vector thuộc tính sang bước 4.

Bước 4: Phân loại hoa.

- Sử dụng phương pháp máy vector hỗ trợ để xây dựng mô hình phân loại.
- Đầu vào của bước này là vector thuộc tính đã qua bước phát hiện hoa.
- Kết quả của mô hình phân loại ở bước này sẽ cho ta danh sách tên các loài hoa theo thứ tự là có độ phù hợp cao nhất đến thấp nhất. Trong ứng dụng này sẽ lấy ra tên 5 loài hoa có độ phù hợp cao nhất.

Bước 5: Tính độ tương tự của các loài.

- Như đã có ở bước thứ 4, kết quả đi ra sẽ là danh sách tên 5 loài hoa. Ở đây, nếu nhiệm vụ của bài toán chỉ là cho ra tên của loài hoa thì có vẻ như không cần bước này. Trong thực tế một loài hoa hồng như có thể xuất hiện trong tự nhiên với khá nhiều màu sắc như hoa hồng đỏ, hoa hồng trắng, vàng. Vì vậy khi tìm kiếm một bông hồng trắng nếu ta sẽ đưa ra danh sách tên của loài hoa đó và kèm theo đó là một bức ảnh có độ tương tự gần giống nhất với bức ảnh cần tìm kiếm của mỗi loài.
- Độ tương tự được tính ở đây là cosin giữa hai vector.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

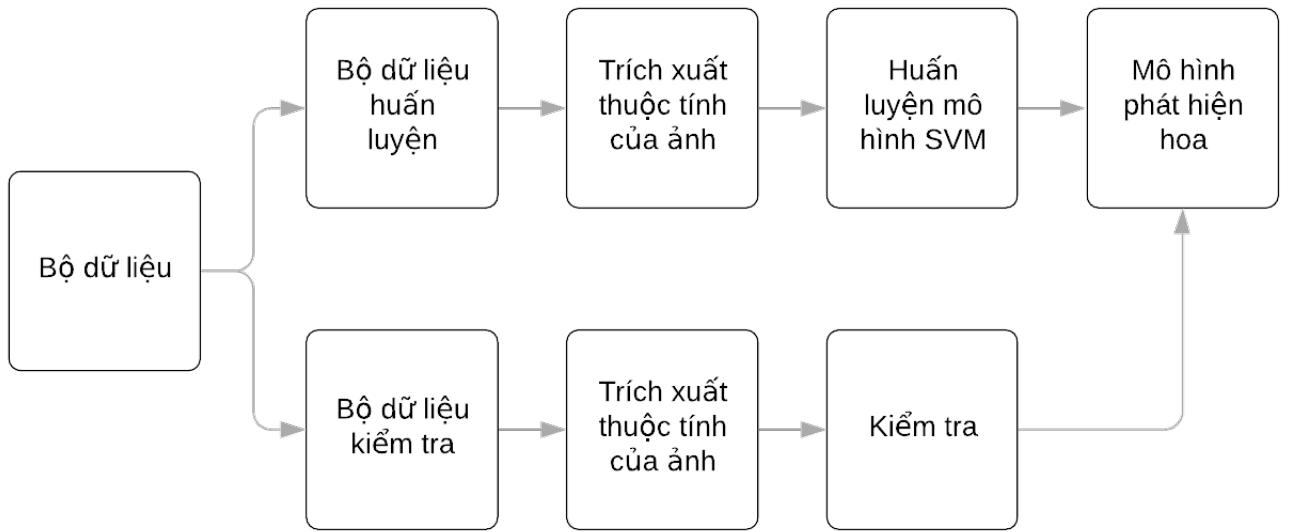
$\|\mathbf{A}\|$ là độ dài của vector thuộc tính A

$\|\mathbf{B}\|$ là độ dài vector thuộc tính B

Độ tương tự sẽ lớn nhất khi vector A trùng vector B và công thức trên cho kết quả bằng 1.

- Trong mỗi loài hoa được nhận ra ảnh nào có độ tương tự lớn nhất với ảnh cần tìm thì ta sẽ hiển thị ra ảnh đó, để người dùng có thể dễ dàng xác nhận được đúng loài hoa cần tìm.

3.2.2 Mô hình phát hiện hoa



Hình 3.4: Mô hình phát hiện hoa

Bước 1: Chuẩn bị dữ liệu.

- Bộ dữ liệu bao gồm 10.290 ảnh trong đó có 5132 ảnh có hoa và 5158 ảnh được thu thập từ bộ dữ liệu ImageNet.
- Chia bộ dữ liệu thành các tập con.
 - Bộ dữ liệu kiểm tra : 1000 ảnh (mỗi lớp có 500 ảnh).
 - Bộ dữ liệu xác nhận : 1000 ảnh (mỗi lớp có 500 ảnh).
 - Bộ dữ liệu huấn luyện: các ảnh còn lại (cụ thể 4132 ảnh có hoa và 4158 ảnh không có hoa).

Bước 2: Trích xuất thuộc tính.

- Ảnh sẽ được thay đổi kích cỡ về dạng 224 x 224.

- Sử dụng mô hình VGG19 [15] để làm bước trích xuất thuộc tính. Chỉ sử dụng đến tầng thứ 19 FC-4096 của mô hình và kết quả thu được ở đây là một vector thuộc tính có 4096 chiều.
- Vector thuộc tính sẽ được chuẩn hóa sẽ được chuẩn hóa L2 (L2 normalization).

Bước 3: Huấn luyện mô hình.

- Sử dụng phương pháp máy vector hỗ trợ để xây dựng mô hình phân loại nhị phân (binary classification).
- Lấy bộ dữ liệu huấn luyện để huấn luyện cho mô hình, và sử dụng bộ xác nhận để đo độ chính xác phân loại của mô hình.
- Kết quả thu được là mô hình phát hiện hoa.

```

1     from sklearn.svm import SVC
2
3     clf = SVC(degree=2, gamma='auto', kernel='linear',)
4     clf.fit(X_train, Y_train)

```

Bước 4: Kiểm tra mô hình.

- Sử dụng bộ kiểm tra để đo độ chính xác của mô hình vừa nhận được.

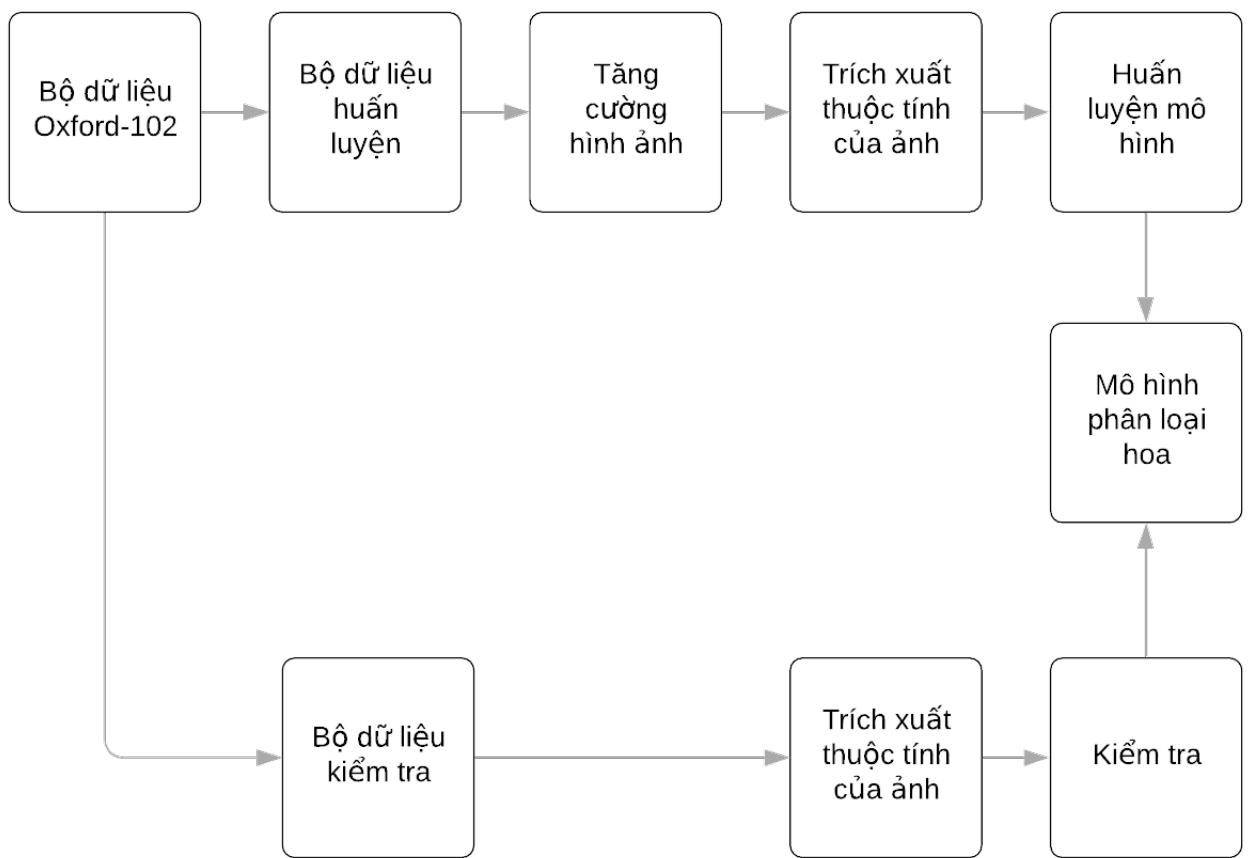
```

1     model = cPickle.load(open(model_name, 'rb'))
2
3     actual_1 = model.predict(flower_validation)
4     actual_0 = model.predict(non_flower_validation)

```

3.2.3 Mô hình phân loại hoa

Bước 1: Chuẩn bị dữ liệu.



Hình 3.5: Mô hình phân loại hoa

- Ở đây sử dụng bộ dữ liệu Oxford-102 [1], nhận thấy số lượng ảnh để huấn luyện cho mô hình chỉ là 10 ảnh mỗi loài hoa, như vậy là hơi ít ảnh vì vậy tôi đã gộp bộ huấn luyện và bộ xác nhận lại thành bộ huấn luyện mở rộng. Sau đó sử dụng phương pháp tăng cường ảnh [16] để tăng số lượng ảnh huấn luyện.

```

1 from keras.preprocessing.image import ImageDataGenerator
2
3 datagen = ImageDataGenerator(
4     vertical_flip=True,
5     horizontal_flip=True,
6 )
  
```

-
- Chia bộ dữ liệu theo cách mới đó là cắt một nửa bộ kiểm tra (mỗi loài hoa trong tập kiểm tra chia một nửa cho bộ huấn luyện nửa còn lại để kiểm tra).
 - Các ảnh trong bộ dữ liệu đều là ảnh chụp cận cảnh bông hoa vì vậy ở mô hình này không cần đến bước tách phông nền.

Bước 2: Trích xuất thuộc tính.

- Đầu vào của bước này đó là bức ảnh mới có trọng tâm là chủ đề cần nhận dạng.
- Ảnh sẽ được thay đổi kích cỡ về dạng 224 x 224.
- Sử dụng mô hình VGG19 [15] để làm bước trích xuất thuộc tính.
- Vector thuộc tính sẽ được chuẩn hóa sẽ được chuẩn hóa L2 (L2 normalization).

Bước 3: Huấn luyện mô hình.

- Sử dụng phương pháp máy vector hỗ trợ để xây dựng mô hình phân loại.
- Lấy bộ dữ liệu huấn luyện để huấn luyện cho mô hình. Kết quả thu được là mô hình phát hiện hoa.

Bước 4: Kiểm tra mô hình.

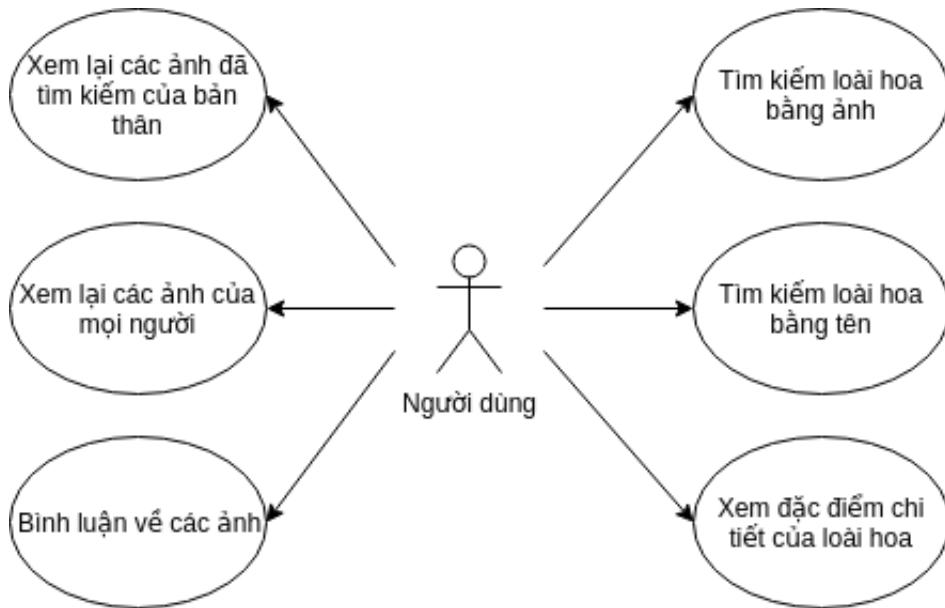
- Sử dụng bộ kiểm tra để đo độ chính xác của mô hình vừa nhận được.

3.3 Xây dựng và cài đặt ứng dụng trên thiết bị di động

Sử dụng bộ thư viện React Native để xây dựng ứng dụng di động. Ưu điểm của React Native để chỉ cần viết một lần duy nhất có thể xây dựng được ứng dụng cho cả iOS và Android.

Ứng dụng sẽ không đơn thuần chỉ có một chức năng tìm kiếm bằng hình ảnh, ngoài ra nó còn có thể được coi là một diễn đàn nơi mọi người có thể chụp lại các bức ảnh về các loài hoa đẹp. Dù khi ứng dụng sẽ gấp phải các loài hoa chưa được thêm vào bộ dữ liệu thì sẽ được mọi người trên diễn đàn trợ giúp.

3.3.1 Mô tả sơ đồ các ca sử dụng chính của ứng dụng



Hình 3.6: Hình ảnh về các ca sử dụng của ứng dụng

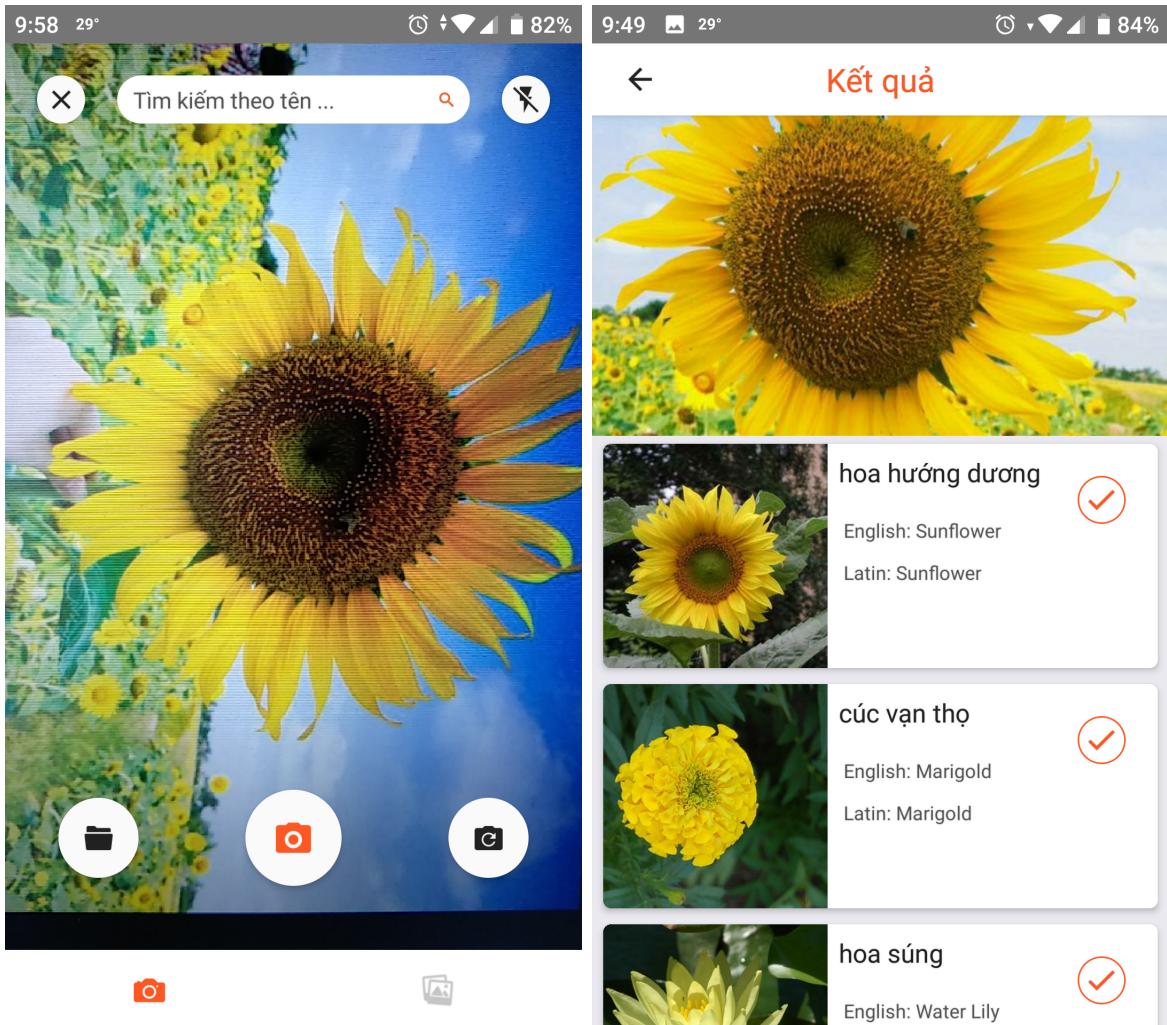
Các ca sử dụng của ứng dụng là :

- Tìm kiếm, nhận dạng loài hoa bằng ảnh: Đầu vào sẽ là một bức ảnh, có thể chụp trực tiếp từ máy ảnh của thiết bị di động hoặc đơn thuần chỉ là một tệp dữ liệu ảnh trong thiết bị di động. Và đầu ra nếu bức ảnh đó không có hoa là thông báo không tìm thấy hoa trong ảnh, hoặc nếu trong ảnh xuất hiện hoa thì là danh sách tên 5 loài hoa phù hợp nhất.
- Tìm kiếm loài hoa bằng tên: Đầu vào sẽ là một chuỗi ký tự nhập từ bàn phím của thiết bị di động, từ khóa sẽ được tìm kiếm và trả về kết quả dựa vào độ trùng khớp với các dữ liệu về tên tiếng Việt, tên tiếng Anh hoặc tên latin.

- Xem đặc điểm chi tiết của loài hoa đó.
- Xem lại các ảnh của bản thân.
- Xem các ảnh của mọi người tải lên hệ thống.
- Bình luận và phản hồi về các ảnh.

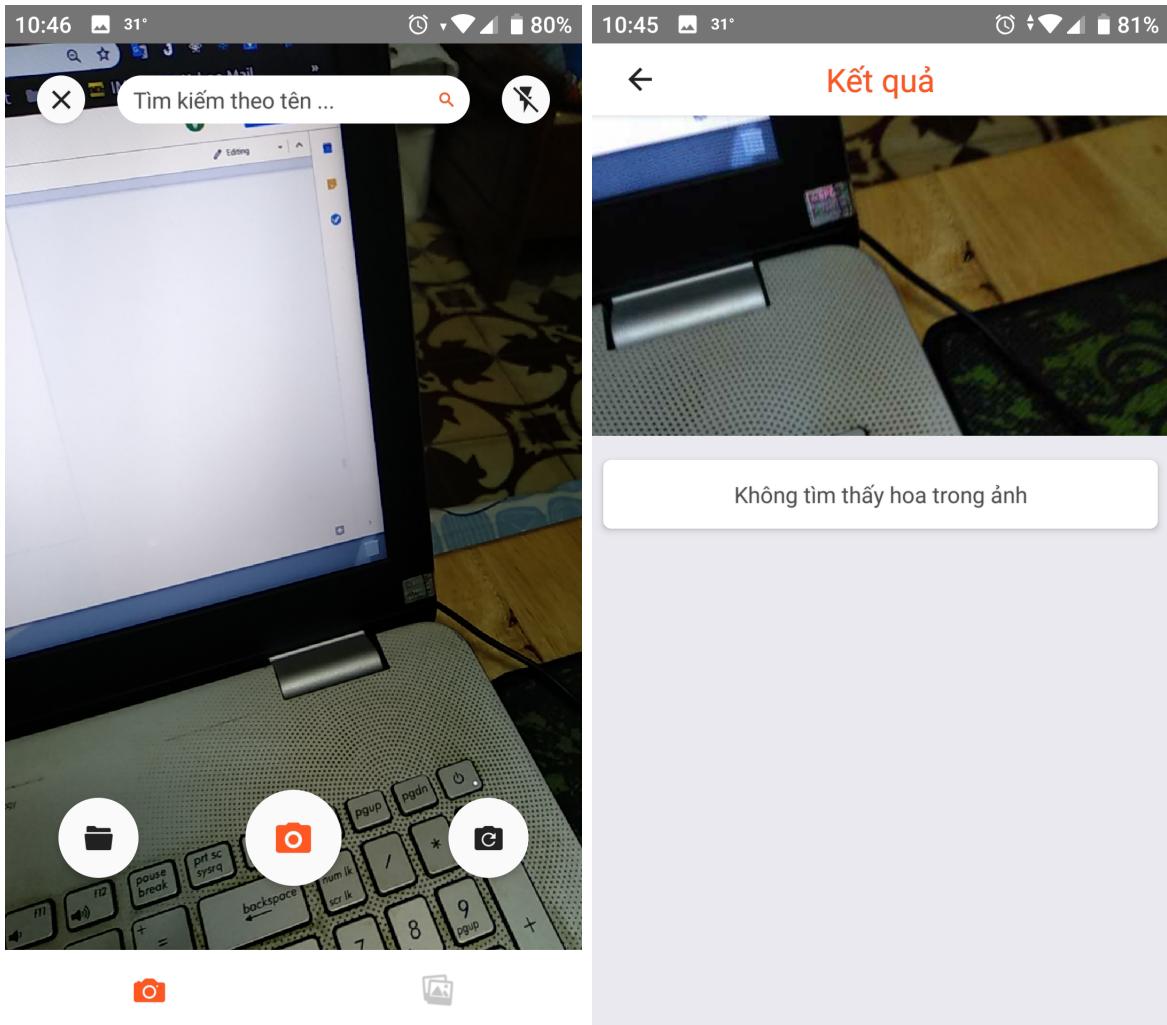
3.3.2 Giao diện của ứng dụng trên điện thoại di động

Ở giao diện tra cứu có các chức năng như là tìm kiếm trực tiếp bằng ảnh chụp hoặc tìm kiếm bằng tên loài hoa. Các chức năng cài đặt cho máy ảnh như tắt/bật đèn chớp, xoay máy ảnh ra phía trước, hoặc sử dụng ảnh có sẵn trong album để tìm kiếm.



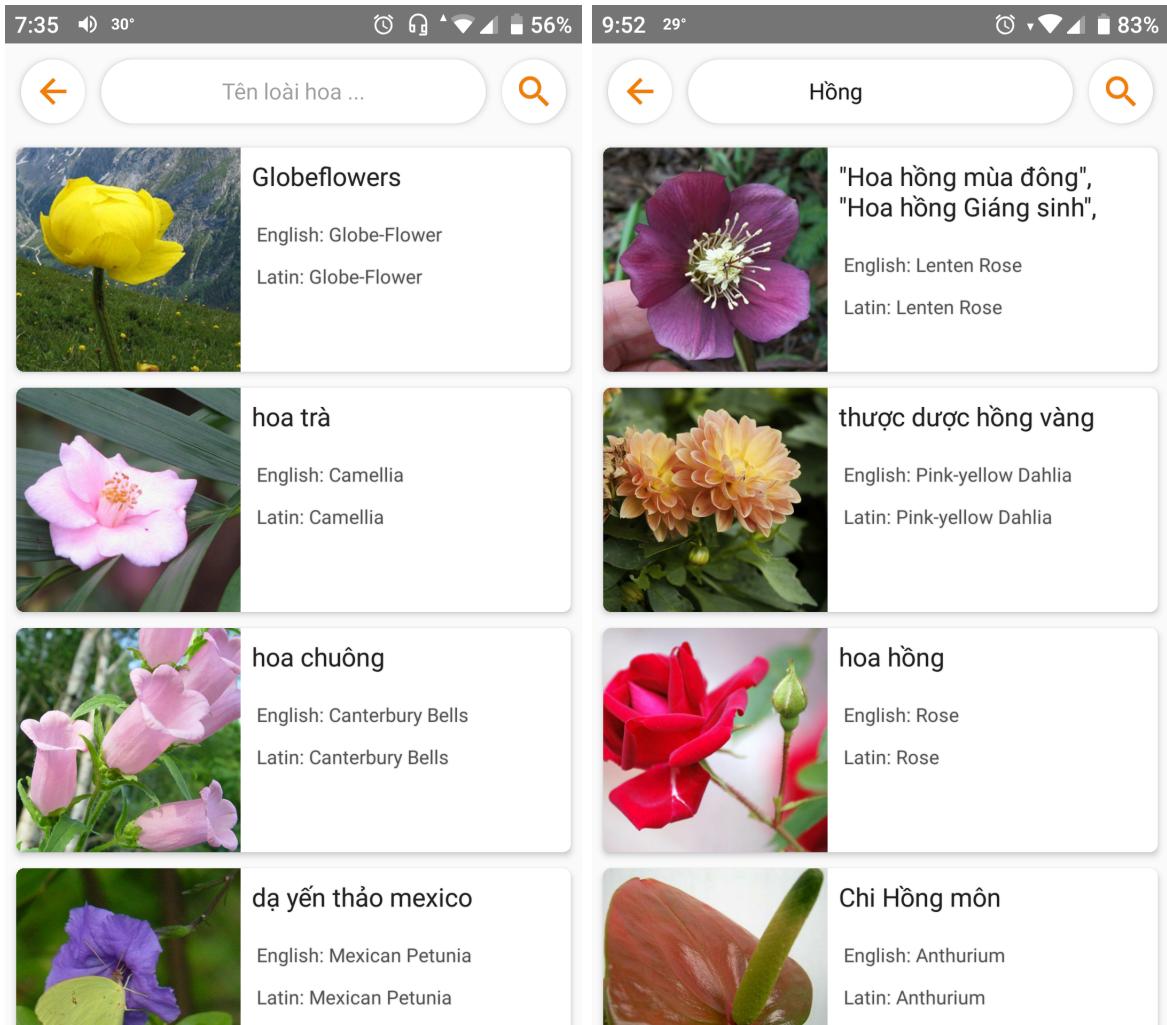
Hình 3.7: Hình ảnh giao diện tra cứu và kết quả tra cứu của ứng dụng.

Ở giao diện kết quả tìm kiếm bằng ảnh ta sẽ nhìn thấy danh sách các loài hoa phù hợp nhất với ảnh chụp đó. Các loài hoa sẽ được sắp xếp theo thứ tự từ trên xuống dưới tương ứng với mức độ phù hợp cao đến thấp. Danh sách tên các loài hoa hiển thị bao gồm tên tiếng Việt, tên tiếng Anh, tên tiếng Latin và đi kèm với đó là hình ảnh đại diện là hình ảnh của loài hoa trong cơ sở dữ liệu có độ tương tự cao nhất với ảnh tìm kiếm.



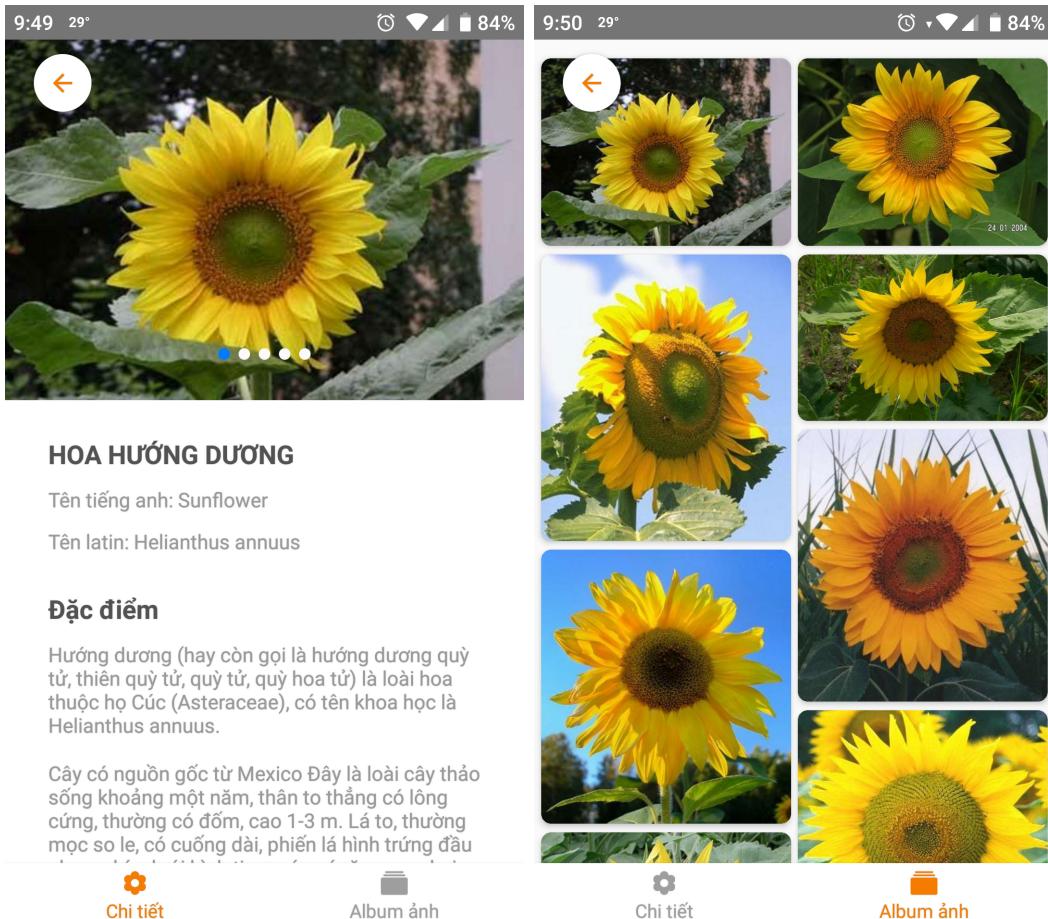
Hình 3.8: Hình ảnh giao diện của ứng dụng khi không phát hiện hoa trong ảnh.

Khi chụp một bức ảnh không có hoa thì kết quả trả về sẽ là không tìm thấy hoa, ở trong hình 3.8 tôi có lấy ví dụ là ứng dụng chụp máy tính của tôi trong hình bên trái và bên phải là ảnh chụp kết quả trả về.



Hình 3.9: Hình ảnh giao diện tìm kiếm bằng tên.

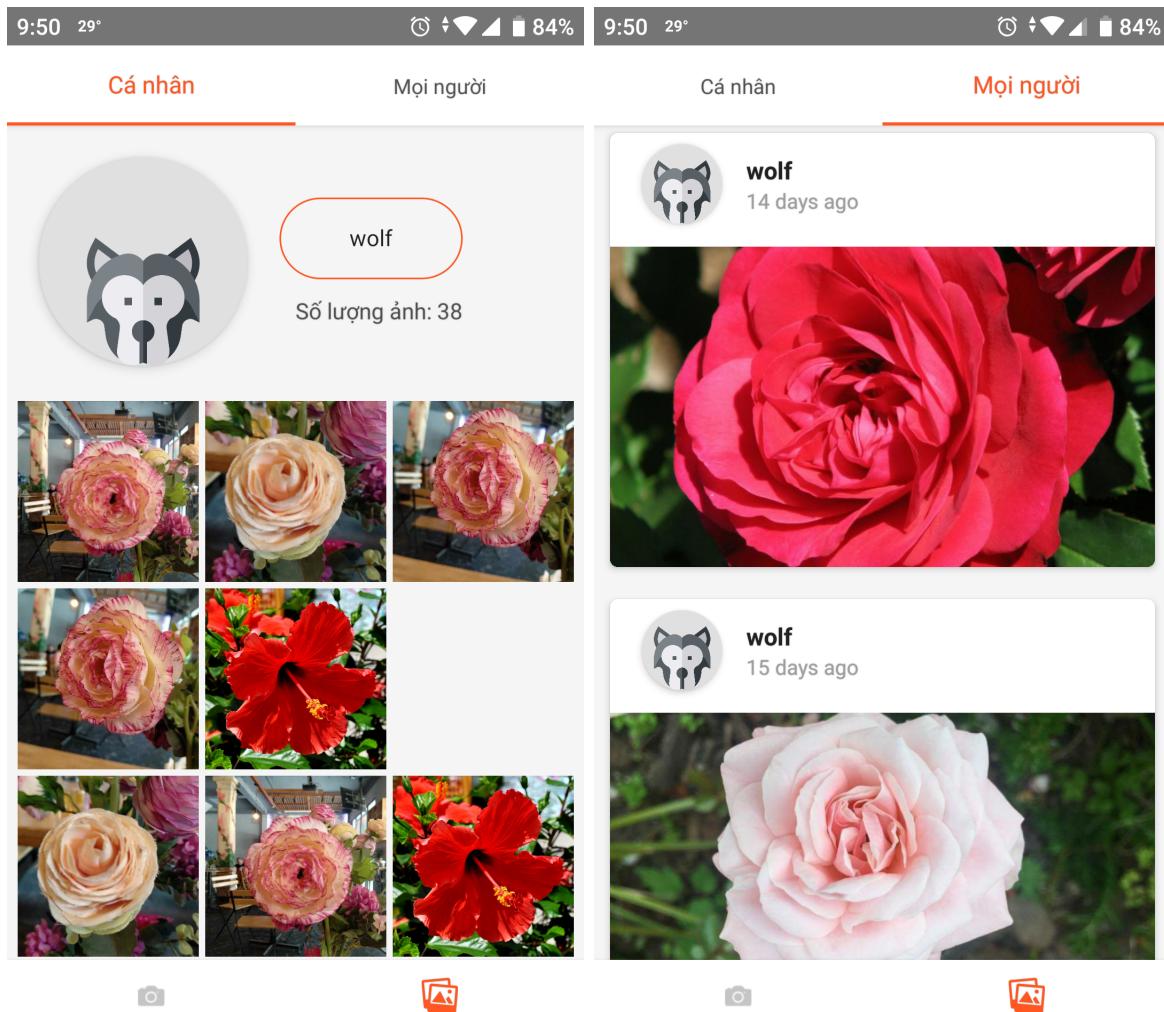
Chức năng tìm kiếm bằng tên sẽ tìm kiếm theo từ khóa mà bạn nhập vào. Từ khóa bạn nhập có thể là tên tiếng Việt, tiếng Anh, hoặc tên tiếng Latin. Ở đây khi bạn chưa nhập bất kỳ từ khóa nào thì ứng dụng sẽ gợi ý danh sách các loài hoa được mọi người tìm kiếm nhiều nhất.



Hình 3.10: Hình ảnh giao diện chi tiết loài hoa.

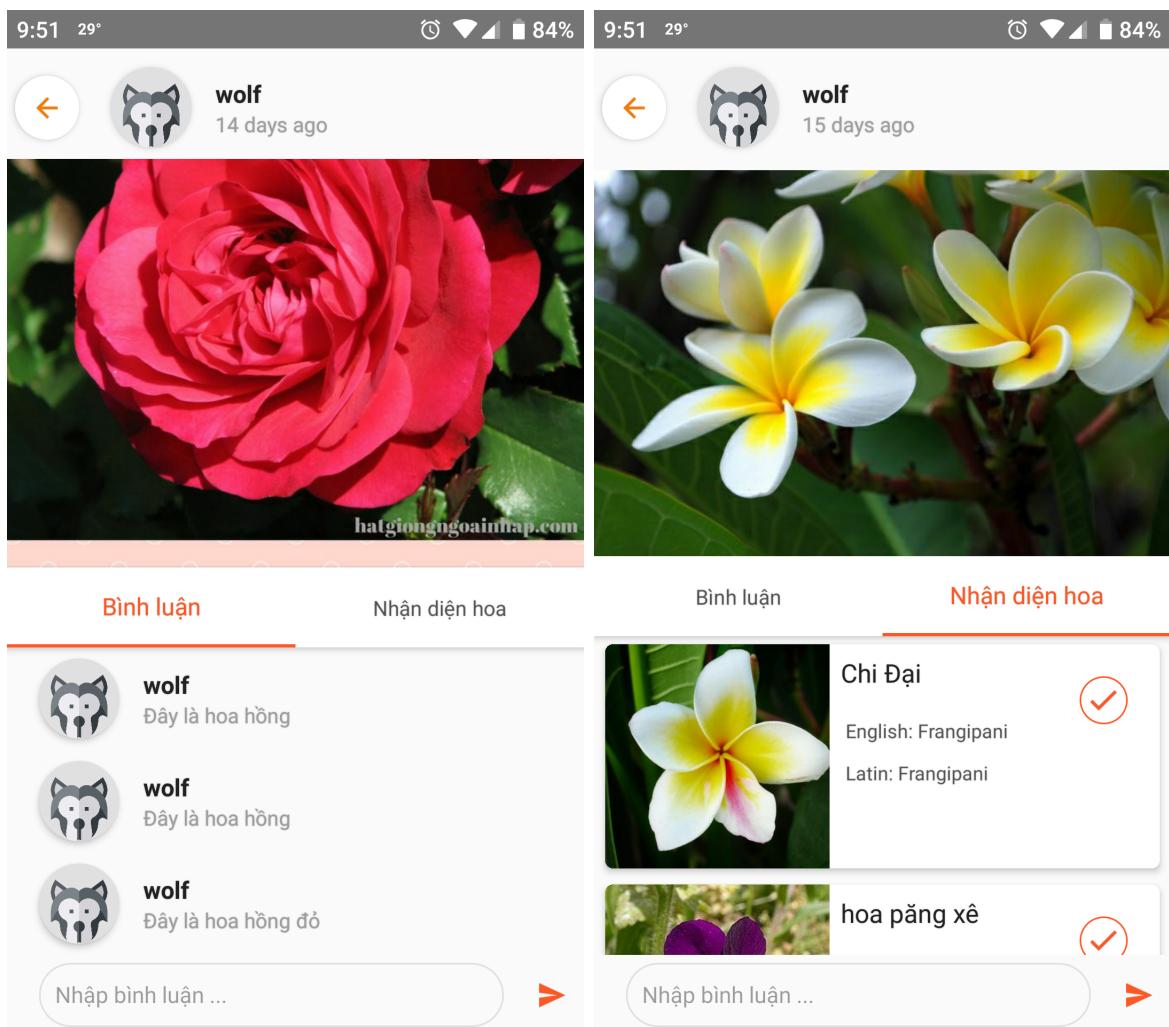
Ở giao diện chi tiết loài hoa tôi có nêu ra tên của loài hoa đó theo tiếng Việt, tiếng Anh, và tên Latin. Đồng thời phía dưới đó chính là đặc điểm của loài hoa đó bao gồm: Đặc điểm chung của loài hoa, đặc điểm sinh trưởng, cách trồng và chăm sóc cây, và ý nghĩa của tên loài hoa. Bên cạnh chức năng xem chi tiết đặc điểm của loài hoa là chức năng xem các ảnh tương tự của loài hoa, các ảnh ở đây là các ảnh trong bộ cơ sở dữ liệu.

Chức năng xem lại các ảnh của bản thân, bạn có thể nhìn thấy danh sách các ảnh mà mình đã từng tìm kiếm. Đồng thời cũng có thể nhìn thấy danh sách các ảnh mà do người dùng khác tìm kiếm.



Hình 3.11: Hình ảnh giao diện xem lại các ảnh của bản thân và mọi người.

Mỗi bức ảnh khi bấm vào ta đều có thể để lại bình luận về loài hoa đó cũng như chia sẻ kinh nghiệm bản thân về cách trồng cũng như chăm sóc loài hoa cho mọi người cùng biết. Đồng thời đưa ra kết quả nhận dạng loài hoa mà ứng dụng đã phản hồi.



Hình 3.12: Hình ảnh giao diện xem nhận xét của mọi người.

Chương 4

Kết quả thực nghiệm

Tại chương 4 này sẽ nói về cách thu thập các bộ dữ liệu, triển khai cài đặt các phần của ứng dụng dựa trên các phương pháp giải quyết vấn đề đã đặt ra ở chương 3. Đồng thời trong chương này cũng công bố các kết quả về các mô hình phân loại đã đạt được.

Chúng ta có một ứng dụng nhận dạng các loài hoa sử dụng thư viện React Native viết cho cả hai hệ điều hành là android và IOS. Ứng dụng sẽ truyền và nhận dữ liệu từ một máy chủ sử dụng các mô hình phân loại đã được huấn luyện.

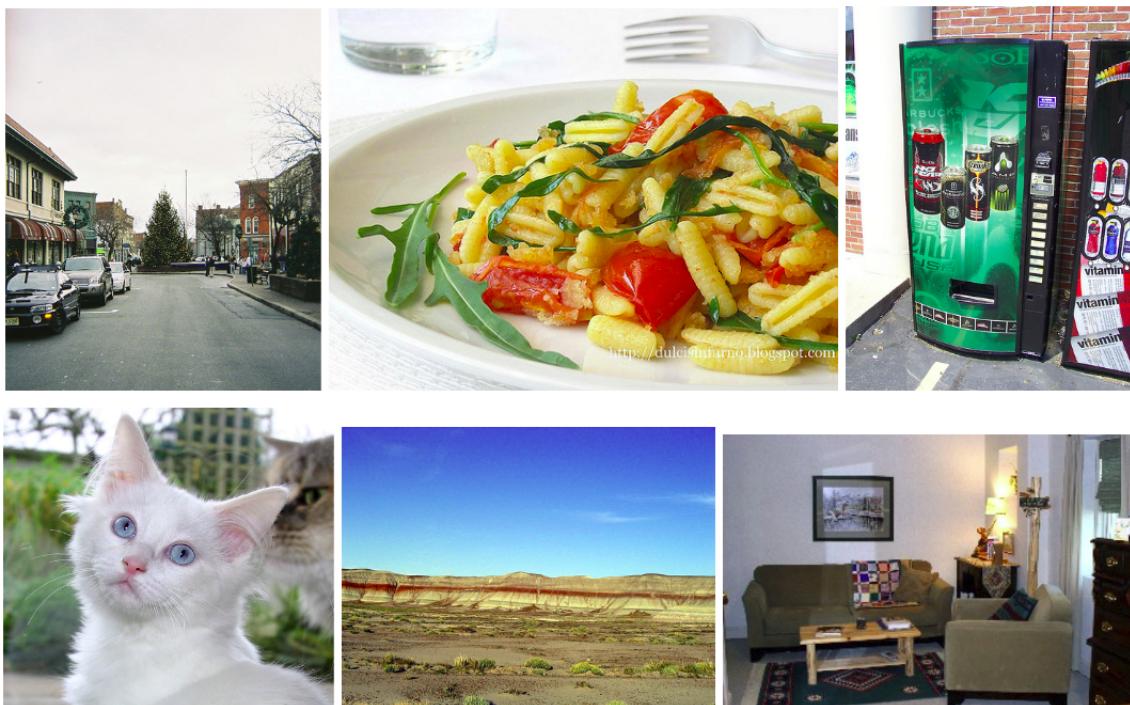
4.1 Xây dựng bộ dữ liệu.

4.1.1 Bộ dữ liệu dùng để phát hiện đối tượng hoa trong ảnh.

Cách thu thập: Sử dụng bộ dữ liệu ImageNet để thu thập các ảnh. Các ảnh tại đây đã được chia sẵn thành các chủ đề nhỏ, đa dạng về chủ đề. Tổng cộng là 5158 ảnh không có hoa. Để lấy các bức ảnh không có hoa tôi lựa chọn các chủ đề:

- Động vật có vú : bao gồm 743 ảnh.
- Cảnh vật trong tự nhiên : bao gồm 930 ảnh.

- Đồ ăn : bao gồm 1063 ảnh.
- Cảnh vật trong phòng : bao gồm 659 ảnh.
- Công cụ máy móc : bao gồm 799 ảnh.
- Đường phố : bao gồm 964 ảnh.



Hình 4.1: Hình ảnh lấy trong bộ ảnh không có hoa

Bộ dữ liệu có hoa tôi lấy từ chủ đề "flower" thu thập được 1207 ảnh. Nhận thấy có sự chênh lệch khá lớn giữa bộ ảnh có hoa và không có hoa nên tôi sử dụng một phần bộ dữ liệu Oxford-102 [1], vào để số lượng ảnh có hoa và không có hoa mang tính cân đối. Tổng cộng là 5132 ảnh trong đó có:

- Ảnh có hoa từ ImageNet: bao gồm 1207 ảnh.
- Ảnh lấy từ Oxford-102 : bao gồm 3925 ảnh.

Cách chia bộ dữ liệu: Bộ dữ liệu được chia thành 3 tập



Hình 4.2: Hình ảnh lấy trong bộ ảnh có hoa

- Tập dữ liệu kiểm tra gồm 1000 ảnh (500 ảnh có hoa và 500 ảnh không có hoa).
- Tập dữ liệu xác nhận gồm 1000 ảnh (500 ảnh có hoa và 500 ảnh không có hoa).
- Tập dữ liệu huấn luyện gồm 8290 ảnh (4132 ảnh có hoa và 4158 ảnh không có hoa).

Khó khăn và hạn chế: Tại trang ImageNet các ảnh để dưới dạng đường link nên khi lọc ảnh đã gặp phải khá nhiều khó khăn như sau.

- Ảnh bị lỗi không thể hiển thị (ví dụ là bộ dữ liệu động vật có vú theo như công bố trên trang web có 1169 ảnh nhưng thực tế khi thu thập chỉ còn 743 ảnh, hay như bộ ảnh hoa công bố là 1924 ảnh nhưng khi thu thập lại chỉ còn 1207 ảnh, các bộ dữ liệu khác cũng tương tự chỉ thu thập được tầm 50-80% ảnh) Đường link không còn được sử dụng, đường link đến hệ thống nước ngoài nên thời gian phản hồi rất lâu
- Một số ảnh quá bé, quá nặng.
- Số ảnh tôi công bố dùng làm bộ dữ liệu ở trên là các ảnh đã bỏ đi các ảnh lỗi ảnh quá bé, quá nhỏ, ...

- Các ảnh trong bộ ảnh không có hoa đôi khi vẫn còn sót lại vài ảnh có xuất hiện bông hoa trong đó. Tương tự như vậy, trong bộ ảnh có hoa vẫn còn có vài ảnh chỉ có chụp thân hoặc gốc cây tỷ lệ này được ghi nhận ở dưới mức 3%. Tôi đã ngồi duyệt lại bằng tay để lọc ra các ảnh đặc biệt.

4.1.2 Bộ dữ liệu dùng để phân loại tên loài hoa

Cách thu thập: Tôi tải về bộ dữ liệu Oxford-102 [1], tại trang web của trường đại học Oxford. Tại bộ dữ liệu có các tập tin đi kèm.

- **imagelabel.mat**: chứa danh sách tên các loài hoa ứng với từng bức ảnh theo dạng số từ 1 tới 102.
- **setid.mat**: chứa cách phân chia bộ dữ liệu ra thành 3 tập dữ liệu nhỏ đó là bộ huấn luyện, bộ xác nhận và bộ kiểm tra.

Chuyển đổi bộ dữ liệu sang tiếng việt. Tại tập tin "imagelabel.mat" của bộ dữ liệu chỉ có tên loài hoa đánh dưới dạng số. Ví dụ trong đó là số 74 tương ứng với loài hoa hồng.

Trong hình 4.3 có thể thấy tiêu đề của trang "Rose (Rosa)" được viết dưới dạng "Tên tiếng Anh(Tên Latin)". Tôi sử dụng công cụ Scrapy dành cho ngôn ngữ python để trích xuất tự động danh sách tên của các loài hoa từ 1 đến 102 vào một file văn bản. Sau đó sử dụng Google và Wikipedia để tra cứu tên tiếng Anh hoặc tên Latin để tìm kiếm ra tên tiếng việt và đặc điểm cũng như cách trồng của từng loại hoa từ đó thêm vào cơ sở dữ liệu.

Trong lúc tìm tên tiếng việt cho các loài hoa tôi gặp phải một số những khó khăn sau:

- Bộ dữ liệu Oxford-102 [1], này bao gồm các loài hoa phổ biến có mọc tại Anh nên một số loài hoa chỉ có giải thích bằng tiếng Anh nên rất khó khăn trong giai đoạn tìm tên, đặc điểm sinh trưởng cũng như cách trồng.
- Khi tìm kiếm trên Google về cách trồng và đặc điểm sinh trưởng tôi phải tìm ở

Rose (Rosa)



Hình 4.3: Hình ảnh về trang web chứa tên loài hoa của bộ dữ liệu Oxford-102

rất nhiều trang web có những cấu trúc HTML rất khác nhau vì vậy công việc này hiện tôi phải làm thủ công bằng tay để có thể tìm được thông tin chi tiết về 102 loài hoa.

4.1.3 Bộ dữ liệu ảnh chụp thực tế.

Cách thu thập.

Mục tiêu của bộ dữ liệu này là dùng để kiểm tra mô hình phân loại. Vì ứng dụng của tôi muốn hướng tới là một ứng dụng nhận dạng hoa dành cho người Việt vì vậy các loài hoa mọi người thường hay tìm kiếm sẽ là các loài hoa phổ biến được trồng và mọc tại Việt Nam. Tôi lựa chọn ở đây gồm có 18 loài hoa phổ biến mọc tại Việt Nam các ảnh ở đây là một phần là do chính tôi chụp bằng điện thoại di động, một phần lấy từ diễn đàn vnphoto.net và google images.

Số lượng ảnh mỗi loài hoa là 40 ảnh mỗi loại vì một loài hoa có thể có nhiều màu ví dụ như hoa giấy có màu tím, hồng và trắng thì số lượng mỗi màu sắc của mỗi loài sẽ gần bằng nhau. Tổng cộng tôi thu thập 720 ảnh của 18 loài hoa danh sách cụ

thể các loài hoa và số lượng màu sắc tối để trong bảng 4.1.

Bảng 4.1: Bảng danh sách các loài hoa phổ biến ở Việt Nam có trong bộ dữ liệu ảnh chụp thực tế

Hoa bìm bìm (hồng, tím)	Hoa giấy (tím, hồng, trắng)
Hoa cẩm chướng (đỏ, vàng, hồng)	Hoa hồng (đỏ, hồng, vàng)
Hoa cúc (cam, vàng, tím)	Hoa hướng dương
Hoa cúc vạn thọ	Hoa ly (cam, vàng)
Hoa đại	Hoa mộc lan
Hoa dâm bụt (đỏ, hồng, vàng)	Hoa păng xê
Hoa đỗ quyên	Hoa sen
Hoa trạng nguyên (đỏ, trắng)	Hoa súng (vàng, trắng)
Hoa trà (trắng hồng)	Hoa thiên điểu

4.2 Môi trường thực nghiệm

Cấu hình phần cứng:

- CPU : Intel core i5-4200H - 2.8Ghz x 4
- RAM : 8GB
- OS : Ubuntu 16.04 LTS - 64 bit

Các thư viện cần cài đặt trước khi tiến hành các thí nghiệm:

- Python 2, NodeJS, Java 8.
- Bộ thư viện sklearn trên python.
- Bộ thư viện keras trên python.

4.3 Kết quả của mô hình phát hiện hoa.

Trong bảng 4.2 là kết quả độ chính xác khi phát hiện các loài hoa.

Bảng 4.2: Bảng kết quả phân loại của mô hình phát hiện hoa

STT	Thuật toán phân loại	Độ chính xác phân loại (%)
1	SVM	98.6

Huấn luyện mô hình

Sử dụng thuật toán SVM được cài đặt sẵn trong bộ thư viện sklearn ở đây tôi thử nghiệm với kernel "linear".

```
1   from sklearn.svm import SVC
2   import numpy as np
3   import cPickle
4   from sklearn.preprocessing import normalize
5
6   def svc_training():
7       X_train = np.concatenate((flower_train, non_flower_train), axis=0)
8       Y_train = load_label(flower_train, non_flower_train)
9       clf = SVC(kernel='linear')
10
11      clf.fit(X_train, Y_train)
12      cPickle.dump(clf, open(filename, 'wb'))
```

Độ đo dùng trong thí nghiệm

Sử dụng độ đo F1-score [21] để đo độ chính xác phân loại của mô hình. Ta coi các ảnh có hoa là 1 và các ảnh không có hoa là 0, các chữ viết tắt được định nghĩa là:

- True Positive-TP: số ảnh của lớp 1 được phân loại đúng là 1.
- True Negative-TN: số ảnh của lớp 0 được phân loại đúng là 0.
- False Positive-FP: số ảnh của lớp 0 bị phân loại nhầm thành 1.

- False Negative-FN: số ảnh của lớp 1 bị phân loại nhầm thành 0.

công thức tính độ đo F1-score [21]

$$F_1 = 2 \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Cài đặt tính toán độ đo F1-score trên python.

```

1 def calcu_f1_score(actual_1, actual_0):
2     TP = np.count_nonzero(actual_1)
3     FP = np.count_nonzero(actual_0)
4     FN = len(actual_1) - TP
5     TN = len(actual_0) - FP
6     Precision = float(TP) / (TP+FP)
7     Recall = float(TP) / (TP+FN)
8     F1_Score = 2*(Recall * Precision) / (Recall + Precision)
9
10    return F1_Score

```

4.4 Kết quả của mô hình phân loại hoa

4.4.1 Kết quả sử dụng bộ test Oxford-102

Trong bảng 4.3 là kết quả độ chính xác khi phân loại hoa. Kết quả tốt nhất đạt được là 78.53% khi sử dụng thuật toán phân loại máy vector hỗ trợ SVM.

Bảng 4.3: Bảng kết quả độ chính xác phân loại hoa

STT	Thuật toán phân loại	Độ chính xác phân loại (%)
1	KNN	61.45
2	Random-forest	65.51
3	SVM	78.53

4.4.2 Giải thích các thí nghiệm

Sử dụng bộ dữ liệu Oxford-102 với tên các bộ dữ liệu được viết tắt:

- data_01: Bộ dữ liệu huấn luyện.
- data_02: Bộ dữ liệu xác nhận.
- data_03: Một nửa bộ dữ liệu kiểm tra.
- data_04: Bộ dữ liệu kiểm tra.

Kết quả khi sử dụng thuật toán KNN

Sử dụng thư viện sklearn có cài đặt sẵn thuật toán k lóng giềng gần nhất KNN.

```
1   from sklearn.neighbors import NearestNeighbors  
2  
3   knn = NearestNeighbors(n_neighbors=1, algorithm='ball_tree')  
4   knn.fit(all_feature_data, all_label)  
5   distances, indices = knn.kneighbors(img_feature_vector)
```

Ở đây độ chính xác của thuật toán được đo bằng kết quả của láng giềng gần nhất ($K=1$). Tôi có sử dụng phương pháp tăng cường ảnh (image augmentation) cụ thể là lật trái, phải, soi vào gương. Như vậy từ một ảnh ban đầu sau phương pháp này ta sẽ có được 4 ảnh. Độ chính xác của thuật toán được kiểm tra trên bộ data_04 như bảng 4.4.

Bảng 4.4: Bảng kết quả phân loại hoa sử dụng phương pháp KNN

STT	Phương pháp	Bộ dữ liệu huấn luyện	Phương pháp tăng cường ảnh	Độ chính xác phân loại (%)
1	KNN	data_01	không	54.65
2		data_01	có	55.04
3		data_01 + data_02	không	60.35
4		data_01 + data_02	có	61.45

Nhận xét:

- Khi sử dụng phương pháp KNN ta sẽ không cần đến thời gian huấn luyện vì đơn giản phương pháp chỉ là tính toán khoảng cách giữa các vector thuộc tính, nhưng thời gian phản hồi của thuật toán tương đối lâu đặc biệt khi số lượng ảnh dùng để huấn luyện tăng lên.
- Độ chính xác của phương pháp KNN sẽ được cải thiện khi ta tăng số lượng ảnh huấn luyện.
- Khi sử dụng cùng một bộ dữ liệu để huấn luyện nhưng khi ta có sử dụng thêm phương pháp tăng cường ảnh thì cũng cải thiện được độ chính xác.
- Tôi nhận thấy kết quả độ chính xác khi sử dụng thêm bộ xác nhận và và phương pháp tăng cường ảnh có cải thiện được kết quả nhưng vẫn tương đối thấp.

Thí nghiệm với thuật toán SVM

Sử dụng thuật toán SVM được cài đặt sẵn trong bộ thư viện sklearn

Độ đo dùng trong thí nghiệm:

Các kết quả được đo bằng hàm score() của thuật toán SVC được cài sẵn trong bộ thư viện sklearn. Hàm score() là trung bình độ chính xác phân loại của các lớp. Trong đó độ chính xác của mỗi lớp là số lượng ảnh phân loại chính xác chia cho tổng số ảnh của mỗi lớp.

huấn luyện mô hình:

```
1   from sklearn.svm import SVC, LinearSVC  
2  
3   model = SVC(degree=3,kernel='poly',decision_function_shape = 'ovr')  
4   model.fit(X_train, Y_train)  
5  
6   Y_test = model.predict(X_test)
```

Trong đó thay đổi các cặp thông số cho thuật toán dựa trên hướng dẫn của bộ thư viện. Các kết quả đạt được tôi công bố ở các bảng phía dưới đây, tại đó chia các kết quả theo từng loại kernel được sử dụng.

Các cặp thông số mặc định của hàm SVC (Support Vector Classification):

SVC(C=1.0, kernel='rbf', degree=3, gamma='auto_deprecated', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None)

Bảng 4.5: Bảng kết quả phân loại hoa sử dụng hàm SVC

STT	Krenel	Bộ dữ liệu huấn luyện	Các thông số cho thuật toán	Dộ chính xác phân loại (%)
5	linear	data_01	kernel='linear', decision_function_shape = 'ovr' (Không tăng cường ảnh)	64.31
6		data_01	kernel='linear', decision_function_shape = 'ovr'	68.29
7		data_01+data_02	kernel='linear', decision_function_shape = 'ovr'	72.29
8		data_01+data_02	kernel='linear', decision_function_shape = 'ovo'	74.5
9	poly	data_01+data_02	degree=3, kernel='poly', decision_function_shape = 'ovr'	63.47
10		data_01+data_02	degree=5, kernel='poly', decision_function_shape = 'ovo'	53.66
11		data_01+data_02	degree=8, kernel='poly', decision_function_shape = 'ovo'	56.75
12	rbf	data_01+data_02	kernel='rbf', decision_function_shape = 'ovr'	61.55
13		data_01+data_02	kernel='rbf', decision_function_shape = 'ovo'	61.55
14	sigmoid	data_01+data_02	kernel='sigmoid', decision_function_shape = 'ovo'	61.55
15		data_01+data_02	kernel='sigmoid', coef0=0.5, decision_function_shape = 'ovo'	61.57

Nhận xét:

- Trong các loại kernel được mà bộ thư viện tôi đã thử cả 4 loại bao gồm: linear, sigmoid, rbf, poly. Kết quả ta có thể thấy như bảng trên trong đó ta nhận thấy kernel linear có kết quả thu được tốt nhất còn lại các loại khác có kết quả gần tương tự nhau và không được cao cho lắm.
- Do kernel linear có kết quả tốt nhất nên tôi sử dụng thêm một hàm nữa có thể tùy biến các thông số tốt hơn dành cho linear đó là linearSVC cũng được cài đặt sẵn trong bộ thư viện sklearn.

```
1 from sklearn.svm import SVC, LinearSVC
2
3 model = LinearSVC(tol=1e-3,C=2.0)
4 model.fit(X_train, Y_train)
5
6 Y_test = model.predict(X_test)
```

Các cặp thông số mặc định của hàm LinearSVC (Linear Support Vector Classification):

LinearSVC(penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1000)

Nhận xét:

- Khi chuyển sang hàm LinearSVC có thể tùy biến nhiều hơn các thông số thu được các kết quả khả quan hơn.
- Kết quả tốt nhất thu được với mô hình số 23 là 78.53 %
- Sử dụng mô dữ liệu huấn luyện mở rộng mà phương pháp tăng cường ảnh làm cho kết quả được cải thiện.

Bảng 4.6: Bảng kết quả phân loại hoa sử dụng hàm LinearSVC

STT	Bộ dữ liệu huấn luyện	Các thông số cho thuật toán	Độ chính xác phân loại (%)
16	data_01	tol=1e-5 (Không tăng cường ảnh)	70.77
17	data_01+data_02	tol=1e-5	78.42
18	data_01+data_02	tol=1e-5, loss='hinge'	76.95
19	data_01+data_02	tol=1e-4	78.42
20	data_01+data_02	tol=1e-3	78.42
21	data_01+data_02	tol=1e-3, loss='hinge'	76.95
22	data_01+data_02	tol=1e-3, loss='hinge', verbose=2	76.95
23	data_01+data_02	tol=1e-3, C=2.0	78.53
24	data_01+data_02	tol=1e-3, C=5.0	78.47
25	data_01+data_02	tol=1e-3, C=100.0	77.5
26	data_01+data_02	tol=1e-3, C=1000.0	77.35

Bảng 4.7: Bảng mô tả các mô hình phân loại bổ sung

STT	Bộ dữ liệu huấn luyện	Các thông số cho thuật toán	Sử dụng tăng cường ảnh
27	data_01+data_02+data_03	tol=1e-3, C=2.0	có
28	data_01+data_02+data_04	tol=1e-5, C=2.0	có

4.5 Kết quả sử dụng bộ test ảnh chụp thực tế

Từ các kết quả khi sử dụng hàm LinearSVC tôi nhận thấy mô hình có số thứ tự 23 cho ta kết quả khả quan nhất 78,53%. Tôi có sử dụng chính mô hình này để thử nghiệm trên bộ dữ liệu ảnh chụp thực tế gồm 18 loài hoa phổ biến tại Việt Nam và thu được các kết quả như bảng 4.8.

Tôi nhận thấy kết quả khi sử dụng mô hình 23 để phân loại các loài hoa trong bộ dữ liệu ảnh chụp thực tế chỉ là 70% vẫn còn tương đối thấp. Vì vậy tôi thử nghiệm thêm 2 mô hình mới dựa trên các phương pháp tăng cường ảnh và thông số cho thuật toán vẫn giống với mô hình 23 nhưng thay đổi bộ dữ liệu huấn luyện. Các mô hình mới được nêu ra trong bảng 4.7.

Kết quả độ chính xác phân loại được báo cáo trong bảng 4.8:

- Ta có thể thấy khi tăng số lượng bộ dữ liệu huấn luyện lên thì độ chính xác phân loại sẽ tăng lên, ở đây đa phần các loài hoa khi chuyển từ mô hình 23 sang mô hình 27 và 28 đều có độ chính xác tăng lên.
- Các loài hoa có độ chính xác rất cao có thể thấy là hoa đai và hoa thiên điểu.
- Các loài hoa dễ bị nhầm lẫn với nhau nhất là hoa sen bị nhầm thành hoa súng, hoa cúc bị nhầm thành hoa hướng dương và ngược lại. Hoa cúc vạn thọ dễ bị nhầm thành hoa cẩm chướng vàng.
- Kết quả độ chính xác trung bình trên các loài hoa đạt được cao nhất với mô hình 28 đó là 78.88%.

Bảng 4.8: Bảng so sánh kết quả phân loại hoa trên bộ ảnh thực tế

Tên loài hoa	Độ chính xác phân loại (%)		
	mô hình 23	mô hình 27	mô hình 28
hoa bìm bìm	90	87.5	92.5
hoa cẩm chướng	67.5	75	75
hoa cúc	67.5	67.5	67.5
hoa cúc vạn thọ	35	42.5	47.5
hoa đai	95	100	100
hoa dâm bụt	57.5	65	67.5
hoa đỗ quyên	37.5	37.5	45
hoa giấy	80	80	82.5
hoa hồng	47.5	95	95
hoa hướng dương	95	97.5	100
hoa ly	62.5	67.5	72.5
hoa mộc lan	90	95	97.5
hoa păng xê	67.5	70	77.5
hoa sen	40	42.5	45
hoa súng	90	95	92.5
hoa thiên điểu	100	100	100
hoa trà	57.5	65	75
hoa trạng nguyên	80	87.5	87.5
Trung bình	70	76.11	78.88

4.6 So sánh kết quả với các ứng dụng đã có sử dụng bộ ảnh chụp thực tế

Từ các kết quả khi sử dụng mô hình bổ sung là mô hình 27 và 28 để phân loại hoa, tôi có so sánh với các ứng dụng đã có và được nêu ở chương 2. Các ứng dụng đã được nêu bao gồm PlantFinder, PlantNet và PictureThis. Với ứng dụng PictureThis số lượng lần tra cứu miễn phí là 5 lần do vậy tôi chỉ có thể so sánh được với 2 ứng dụng còn lại là PlantFinder và PlantNet.

Sử dụng mô hình phân loại tốt nhất là mô hình 28 với các ứng dụng là PlantFinder và PlantNet. Bộ dữ liệu sử dụng để kiểm tra là bộ ảnh chụp thực tế gồm 720 ảnh thuộc 18 loài hoa phổ biến ở Việt Nam. Độ chính xác của các ứng dụng khác được tính là kết quả phân loại đầu tiên, các kết quả này đang là tiếng Anh do vậy cần phải dịch sang tiếng Việt. Kết quả được nêu ra trong bảng 4.9

Nhận xét:

- Với ứng dụng PlantFinder ta có thể thấy sự khác biệt trong các kết quả độ chính xác, một số loài có độ chính xác rất cao vì dụ như hoa đỗ quyên hoặc hoa trà nhưng đa phần là không nhận dạng được các loài hoa như hoa bìm bìm hoặc hoa cẩm chướng. Có thể bộ dữ liệu về hoa của ứng dụng chưa đầy đủ và thiếu các loài hoa phổ biến tại Việt Nam.
- Với ứng dụng PlantNet các loài hoa đa phần có độ chính xác tốt, có thể kể đến như hoa bìm bìm hoặc hoa hồng đặc trưng và có độ chính xác 100%.

Bảng 4.9: Bảng so sánh kết quả phân loại hoa của các ứng dụng sử dụng bộ ảnh chụp thực tế

Tên loài hoa	Độ chính xác phân loại (%)		
	PlantNet	PlantFinder	mô hình 28
hoa bìm bìm	100	0	92.5
hoa cẩm chướng	95	0	75
hoa cúc	80	25	67.5
hoa cúc vạn thọ	60	0	47.5
hoa đại	100	0	100
hoa dâm bụt	87.5	0	67.5
hoa đỗ quyên	70	87.5	45
hoa giấy	75	55	82.5
hoa hồng	100	72.5	95
hoa hướng dương	62.5	85	100
hoa ly	87.5	75	72.5
hoa mộc lan	80	67.5	97.5
hoa păng xê	72.5	85	77.5
hoa sen	90	17.5	45
hoa súng	52.5	30	92.5
hoa thiên điểu	95	0	100
hoa trà	60	85	75
hoa trạng nguyên	95	0	87.5
Trung bình	81.25	38.05	78.88

4.7 So sánh kết quả tìm kiếm dùng Google Image

Một thống kê khác cũng khá quan trọng đó chính là tìm kiếm bộ ảnh chụp thực tế với chính Google Image. Kết quả tìm kiếm được thể hiện trong bảng 4.10

Nhận xét:

- Tỷ lệ về kết quả phản hồi được hiển thị bằng tiếng Anh là 98.9% ngoài ra là các thứ tiếng như : Ấn Độ, Nga, Tây Ban Nha, Trung Quốc, Đức, Indonesia. Và đặc biệt không có một kết quả nào được hiển thị với tiếng Việt.
- Tỷ lệ kết quả nhận dạng sai trung bình khá thấp nhưng có một số trường hợp đặc biệt ở đây ví dụ như: Hoa cúc hay bị nhầm lẫn với hoa hướng dương, Hoa sen hay bị nhầm với hoa súng. Hoa cúc vạn thọ hay bị nhầm với hoa cẩm tú cầu. Hoa súng hay bị nhầm với hoa sen.
- Tỷ lệ kết quả trả về là không liên quan đến tên loài hoa là một vấn đề trước đây không được mấy quan tâm, các kết quả không liên quan thuộc nhiều chủ đề khác nhau như tên công ty bán hoa, tên ngày lễ hội ở các nước...

Bảng 4.10: Bảng so sánh kết quả tìm kiếm bằng Google Image

Tên loài hoa	Tỷ lệ kết quả trả về (%)			
	Tiếng Việt	Tiếng Anh	Nhận dạng sai	Không liên quan
hoa bìm bìm	0	97.5	22.5	5
hoa cẩm chướng	0	100	12.5	7.5
hoa cúc	0	100	32.5	15
hoa cúc vạn thọ	0	100	12.5	2.5
hoa đai	0	97.5	7.5	2.5
hoa dâm bụt	0	97.5	5	0
hoa đỗ quyên	0	97.5	7.5	0
hoa giấy	0	100	0	0
hoa hồng	0	97.5	2.5	0
hoa hướng dương	0	100	2.5	0
hoa ly	0	100	0	0
hoa mộc lan	0	100	0	0
hoa păng xê	0	100	0	0
hoa sen	0	97.5	7.5	0
hoa súng	0	95	15	2.5
hoa thiên điểu	0	100	0	0
hoa trà	0	100	15	5
hoa trạng nguyên	0	100	5	2.5
Trung bình	0	98.9	8.19	2.78

4.8 Thảo luận về các kết quả

Những trường hợp đã được giải quyết

Những trường hợp đã nhận dạng chính xác và cho kết quả nhận dạng là chính xác là những ảnh chụp đơn bông hoa, ảnh được chụp cận cảnh, không quá tối, chủ thể hoa không quá bé và ảnh có độ nhiễu thấp.



Hình 4.4: Hình ảnh ví dụ trường hợp nhận dạng đúng

Những trường hợp còn nhận dạng sai Trong mô hình phân loại còn gặp phải khá nhiều trường hợp phân loại sai do một số nguyên nhân sau đây.

- Ảnh quá tối, hoặc bị nhiễu nhiều.
- Bộ dữ liệu huấn luyện Oxford-102 chứa các ảnh chụp cận cảnh và đơn lẻ của mỗi loài hoa nhưng trong thực tế có khá nhiều hoa nở thành từng chùm ví dụ điển hình ở đây đó là hoa giấy. Người dùng đôi khi gặp khó khăn khi tách từng bông hoa riêng lẻ ra để nhận dạng. Và như vậy dẫn đến trường hợp nhận dạng sai.

- Trong ảnh có thể xuất hiện nhiều hơn một loài hoa. Ví dụ cụ thể như một lồng hoa thường được cài lên rất nhiều loài hoa để tăng tính thẩm mỹ. Hiện tại hệ thống tách phông nền mới đang tách ra chủ thể có kích thước lớn nhất. Nhưng không phải lúc nào người dùng cũng muốn chỉ nhận biết tên một loài hoa mà còn phải bao gồm các hoa trong ảnh.



Hình 4.5: Hình ảnh ví dụ về trường hợp nhận dạng sai

Chương 5

Kết luận

Trong khóa luận này trình bày một ứng dụng nhận dạng tên loài hoa dựa vào ảnh đầu vào trên nền tảng di động. Ứng dụng này được xây dựng với ba mô hình học máy chính: (1) một mô hình phân loại nhị phân để phát hiện có đối tượng hoa trong ảnh hay không, (2) một mô hình nhận dạng tên của loài hoa trong ảnh, và (3) một thuật toán tìm kiếm ảnh tương tự khi hiển thị các ảnh kết quả.

Ứng dụng được phát triển sử dụng phương pháp sử dụng lại một mô hình học sâu đã được huấn luyện trên bộ dữ liệu lớn khác để trích xuất dữ liệu của ảnh VGG19 [15]. Thực nghiệm so sánh phương pháp tiếp cận này với các cách trích xuất thuộc tính truyền thống chỉ ra tính khả thi và tiết kiệm của việc sử dụng lại mô hình giữa các bài toán nghiên cứu trong lĩnh vực xử lý ảnh.

Những đóng góp chính của khóa luận là: (1) Việt hoá tên cũng như đặc điểm về sinh trưởng, cách trồng của 102 loài hoa trong bộ dữ liệu Oxford-102, (2) phát triển ứng dụng trên nền tảng di động, và (3) phát triển mô hình phân loại ảnh có hoa hay không với độ chính xác 98.6% và mô hình nhận dạng tên loài hoa với độ chính xác là 78.56%.

Hướng phát triển tiếp theo của nghiên cứu là tìm cách khắc phục các điểm còn hạn chế trong việc nhận dạng hoa như nhận dạng đa chủ thể hoa hay nhận dạng hoa khi mọc cả chùm.

Tài liệu tham khảo

Tiếng Anh

- [1] Nilsback, M-E., and Andrew Zisserman. "A visual vocabulary for flower classification." *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE, 2006.
- [2] Nilsback, Maria-Elena, and Andrew Zisserman. "Automated flower classification over a large number of classes." *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008.
- [3] Xie, Lingxi, et al. "Image classification and retrieval are one." *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. Acm, 2015.
- [4] Sharif Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014.
- [5] Varma, Manik, and Debajyoti Ray. "Learning the discriminative power-invariance trade-off." *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007.
- [6] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.

- [7] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *international Conference on computer vision & Pattern Recognition* (CVPR'05). Vol. 1. IEEE Computer Society, 2005.
- [8] Boykov, Yuri Y., and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." *Proceedings eighth IEEE international conference on computer vision*. ICCV 2001. Vol. 1. IEEE, 2001.
- [9] Bertelli, Luca, et al. "Kernelized structural SVM learning for supervised object segmentation." *CVPR* 2011. IEEE, 2011.
- [10] Perez, Luis, and Jason Wang. "The effectiveness of data augmentation in image classification using deep learning." *arXiv preprint arXiv:1712.04621* (2017).
- [11] Varma, Manik, and Andrew Zisserman. "Classifying images of materials: Achieving viewpoint and illumination independence." *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2002.
- [12] Alexe, Bogdan, Thomas Deselaers, and Vittorio Ferrari. "Measuring the objectness of image windows." *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012): 2189-2202.
- [13] Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2), 154-171.
- [14] Cheng, M. M., Zhang, Z., Lin, W. Y., & Torr, P. (2014). BING: Binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3286-3293).
- [15] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
- [16] Bloice, Marcus D., Christof Stocker, and Andreas Holzinger. "Augmentor: an image augmentation library for machine learning." *arXiv preprint arXiv:1708.04680* (2017).

- [17] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248-255. Ieee, 2009.
- [18] Longman, Ram. "Embedded CycleGAN for Shape-Agnostic Image-to-Image Translation." (2018).
- [19] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. "Imagenet large scale visual recognition challenge." *International journal of computer vision* 115, no. 3 (2015): 211-252.
- [20] Retrieval, Image. "Current Techniques, Promising Directions, and Open Issues" Yong Rui and Thomas S." *Huang Department of ECE & Beckman Institute, University of Illinois.*
- [21] Sokolova, Marina, and Guy Lapalme. "A systematic analysis of performance measures for classification tasks." *Information Processing & Management* 45.4 (2009): 427-437.